SHARP
SOFTWARE
MANUAL

X68030用

Human68k ver 3.0

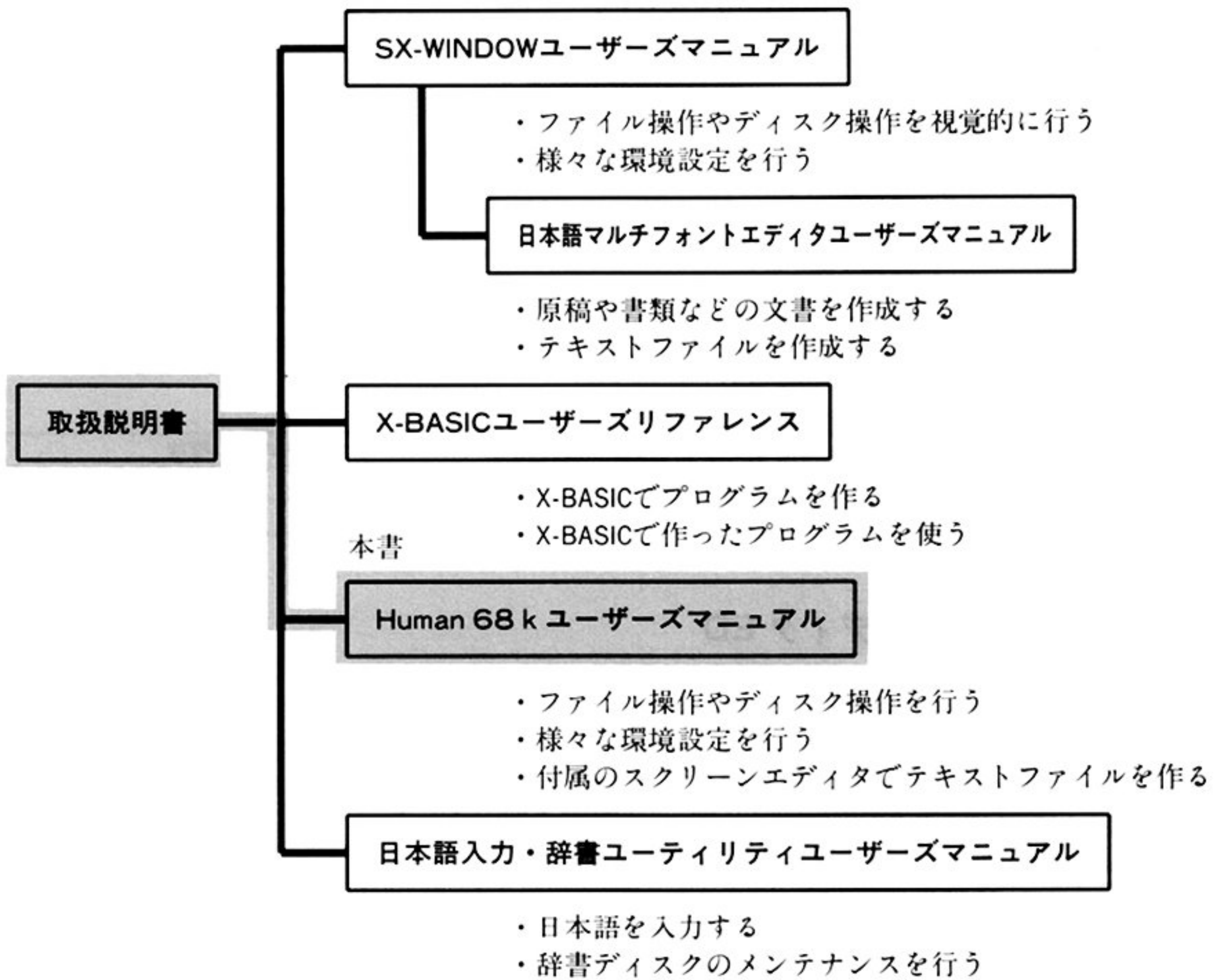
ユーザーズマニュアル

Human68k
ユーザーズ マニュアル

はじめに

本書は、シャープパーソナルワークステーション「X68030シリーズ」上で動作するオペレーティングシステム「Human68k」の説明書です。Human68k の機能、およびその使い方が詳細に解説されています。

なお、本機の取り扱いについては、別冊の「取扱説明書」にわかりやすく説明されています。初めて本機をお使いになる方は、必ず「取扱説明書」を先にご覧ください。

下図は、本機に同梱の6冊のマニュアルの互いの関係を示しています。

マニュアルの使い方

このマニュアルは、初めてお使いになる方はもちろん、ひととおりHuman68kを理解した方でも使えるようになっております。

初めての方は、まず第1部第1章から4章までご覧いただき、オペレーティングシステムやHuman68kの概要などを理解されるよう、お勧めします。このあと、使ってみたいコマンドを選び、まず具体例を参考にさわってみたり、付属のスクリーンエディタの使い方へと発展するとよいでしょう。

ものは試しです。システムディスクのバックアップをとり、キーを押しながらいろいろと試してください。まちがいを恐れずチャレンジしてください。

ひととおりHuman68kを理解された方は、辞書代わりとして活用していただくと、コマンドの書式や機能などをより深く理解されるはずです。

また、目次や索引も、本書の構成にそった〈目次〉、コマンド名から目的のページや機能をさがせる〈コマンド別目次〉、お使いになる目的に応じて目的のページやコマンドをさがせる〈索引〉、と使い方に応じて構成されています。大いにご活用ください。

各部の構成は、次のとおりです。

第1部　Human68k

各　章	内　　　容
第1章	オペレーティングシステムの概要や、Human68kの構成および機能の概略など
第2章	Human68kの起動から終了の方法まで、フロッピーディスクのフォーマットやバックアップコピーの作成
第3章	Human68kにおけるファイルとディレクトリの扱い
第4章	Human68kで使うことができるコマンドの概略とその種類など、バッチ処理やリダイレクト機能
第5章	各コマンドの詳細な使い方 コマンドはアルファベット順に並んでいます
第6章	テンプレート機能やヒストリ機能などのキー入力の編集機能、キーボードコントロール機能
第7章	CONFIG. SYSファイルによるシステム構築の方法
第8章	デバイスエラーの処置方法

第2部　スクリーンエディタED

各　章	内　　　容
第1章	スクリーンエディタＥＤの機能の概略、起動方法と終了方法など
第2章	実際の編集作業で使うEDのコマンドの使い方 機能別に分類されています
第3章	EDをより高度に使うための、スイッチをつけた起動方法
第4章	EDの簡単な操作例
第5章	EDのコマンドやキー操作の一覧

付　録

ASCII制御コード、エスケープシーケンスの一覧など、Human68kを使う上で必要な資料

本書で用いる表記法

Human68k のコマンドの書式は、以下の表記法に従って表現しています。

その他カンマ（ , ）、コロン（：）、スラッシュ（／）、等号（＝）などの記号は表記されている通りに入力してください。入力する位置も変えてはいけません。

〔 〕	**角形カッコ**	オプションです。必要に応じて入力してください。
〈 〉	**山形カッコ**	あなたが入力するデータです。 たとえば次のような場合は、〝PATH〟に続けて実際の〝パス名〟を入力します。 **PATH 〈パス名〉**
｛ ｝	**波形カッコ**	カッコの中から必要な項目を選んで入力します。 角形カッコで囲まれていない限りは、最低１つの入力が必要です。
｜	**縦　線**	下記のように、選択項目の区切りに使います。 **〈パス名〉 ｜ 〈ファイル名〉** ただし、フィルタの中で使用した場合は、パイプを表します（４．７．２項参照）。 **DIR ｜ SORT（この場合、｜はパイプ）**
……	**省略記号**	オプションです。必要に応じて数度繰り返して入力する項目です。
[　　]	**ケイ囲み**	大文字が[　　]で囲まれているときは、特殊な文字の入力を表します。 [CTRL]＋[C]などはコントロール文字を表します。
[↵]	**リターンキー**	リターンキーの入力を表します。

なお、本書中でコマンド入力の例をあげる際には、〝A>〟などのプロンプト記号を省略し、キー入力すべき文字を大文字によって表記してあります。

目　次

第1部　Human68k

第1章　Human68kの概要　3

1.1　Human68kとは …… 3
1.2　Human68kの特長 …… 4
1.2.1　バックグラウンド処理 …… 4
1.2.2　ネットワーク環境への対応 …… 5
1.2.3　ヒストリデバイスドライバ …… 5
1.2.4　ファイルアクセス …… 6
1.2.5　対話型メニューコマンドの採用 …… 6
1.2.6　ハードディスクのパーティション(領域分割) …… 6
1.3　Human68k ver.3.0の特長 …… 7
1.3.1　CPU種類の判別 …… 7
1.3.2　SCSIデバイスドライバの標準装備 …… 7
1.3.3　CONFIGファイル機能の充実 …… 7
1.3.4　MIDI対応、OPMDRV3.Xの搭載 …… 7

第2章　Human68kの起動　9

2.1　イントロダクション …… 9
2.2　Human68kの起動 …… 9
2.3　システム終了の方法 …… 11
2.4　フロッピーディスクのフォーマット …… 11
2.5　フロッピーディスクのバックアップ …… 13
2.6　自動実行バッチファイル(AUTOEXEC.BAT)について …… 16
2.7　ファイル …… 16
2.8　内蔵ハードディスクの初期化と領域確保 …… 17
2.9　システム、辞書ディスクの内容を内蔵ハードディスクに転送する(内蔵ハードディスクからの起動) …… 20

第3章　ファイルとディレクトリ　23

3.1　イントロダクション……23
3.2　ファイル名の付け方……23
3.3　ワイルドカード……24
3.4　予約ファイル名……25
3.5　ファイルの取り扱い……26
3.5.1　ファイルのコピー……26
3.5.2　ファイルの作成……26
3.5.3　ファイルの削除……27
3.6　ディレクトリ……28
3.7　ディレクトリの取り扱い……30
3.7.1　パス名……30
3.7.2　ディレクトリの移動……32
3.7.3　ディレクトリの作成……32
3.7.4　ディレクトリの削除……33

第4章　コマンドについて　35

4.1　イントロダクション……35
4.2　コマンドの種類……35
4.3　コマンドオプション……36
4.4　コマンドの共通事項……37
4.5　外部コマンドとPATH……37
4.6　バッチ処理……38
4.6.1　パラメータを用いたバッチファイルの作り方……40
4.6.2　自動実行バッチファイル(AUTOEXEC. BAT)の作り方……41
4.7　コマンド行で利用できる機能……42
4.7.1　リダイレクト機能……42
4.7.2　パイプ機能……43
4.7.3　マルチ処理機能……44

第5章　コマンド　45

5.1　イントロダクション……45
5.2　Human68kのコマンド……45

ATTRIB……48
BACKUP……50
BIND……52
BREAK……55
CACHE……56
CHDIR(CD)……57
CHKDSK……58
CLS……60
COMMAND……61
COPY……64
COPY2……68
COPYALL……72

CTTY …………………… 74
DATE …………………… 75
DEL(ERASE) …………… 77
DIR………………………79
DISKCOPY ……………… 82
DRIVE …………………… 88
DUMP …………………… 90
EXIT………………………91
FASTIO ………………… 92
FASTOPEN………………93
FASTSEEK……………… 94
FC ………………………… 95
FIND……………………… 97
FORMAT ………………… 99
HIS ……………………… 108
KEY……………………… 110
MEMFREE …………… 114
MKDIR(MD) …………… 115
MORE…………………… 116
MOVE…………………… 117
PATH…………………… 118
PR ……………………… 120
PROCESS………………122
PROMPT…………………125
RECOVER ………………127
REN(RENAME)…………130
RESTORE ………………132
RMDIR(RD)………………134
SCREEN ……………… 135
SET ………………………137
SORT…………………… 139
SPEED………………… 140
SUBST………………… 144
SWITCH……………… 146
SYS ………………………161
TEMP ………………… 163
TIME …………………… 164
TIMER………………… 165
TREE …………………… 168
TYPE …………………… 169
USKCGM …………… 170
VER……………………… 175
VERIFY………………… 176
VOL……………………… 177
WHERE………………… 179

5．3　バッチ処理コマンド …………………………………………… 180

ECHO …………………… 181
FOR ……………………… 183
GOTO …………………… 184
IF ………………………… 185
PAUSE ………………… 186
REM……………………… 187
SHIFT…………………… 188

第6章　キー入力の編集とキーボードコントロール　189

6．1　イントロダクション ……………………………………………189
6．2　テンプレート機能 ………………………………………………189
6．3　ヒストリ機能(HISコマンド) ………………………………… 194
6．4　ヒストリデバイスドライバ………………………………………197
6．4．1　HISTORY.Xの概要 …………………………………………197
1．コマンド行での編集機能………………………………………… 197
2．テンプレートの操作……………………………………………… 197
3．入力文字列のヒストリの検索と再入力(ヒストリ機能) ……… 197
4．ディレクトリ変更履歴の検索と再入力 …………………………198
5．エイリアス(別名定義)機能……………………………………… 198

6．簡易バッチ機能 ……199
7．環境変数の取り込み・その他 ……199

6．4．2　HISTORY.Xの機能 ……199
1．コマンド行での編集機能 ……200
2．テンプレートの操作 ……202
3．入力文字列のヒストリの検索と再入力(ヒストリ機能) ……203
4．ディレクトリ変更履歴の検索と再入力 ……206
5．エイリアス(別名定義)機能 ……207
6．簡易バッチ機能 ……209
7．環境変数の取り込み・その他 ……209

6．4．3　定義ファイルについて ……210
1．ヒストリ定義ファイル ……211
2．キー定義ファイル ……211

6．4．4　オプションについて ……212
1．組み込み時(初期起動時)に設定できるもの ……213
2．随時設定変更できるもの(初期組み込み時にも設定可能) ……214
3．組み込み時以外にのみ設定できるもの ……216

6．4．5　HISTORY.Xの運用について ……218
1．エイリアス機能の利用 ……218
2．ヘルプファイルについて ……221
6．5　キーボードコントロール機能 ……222

第7章　システムの構築　223

7．1　イントロダクション ……223
7．2　CONFIG.SYSファイルのコマンド ……223
BELL ……226
BREAK ……226
BUFFERS ……227
COMMON ……227
DEVICE ……228
DIRSCH ……231
ENVSET ……231
EXCONFIG(CONFIGED) ……232
FILES ……235
KEY ……235
LASTDRIVE ……236
PROCESS ……236
PROGRAM ……237
SCSIDEV ……237
SHARE ……238
SHELL ……238
TITLE ……239
USKCG ……239
VERIFY ……240

7.3　FDDEVICEドライバについて……241
7.4　OPMドライバについて……242
7.4.1　音源の制御……245
7.4.2　FM音源の音色データ(レジスタ)……259
7.5　プリンタ設定について……267
7.6　RAMディスクについて……268
第8章　デバイスエラーの処置方法　270
8.1　イントロダクション……270
8.2　デバイスエラーメッセージ……270
8.3　デバイスエラーの処置方法……270

第2部　スクリーンエディタED

第1章　スクリーンエディタEDの概要　275

1.1　EDとは　275
1.2　EDの機能　275
1.3　EDの起動　275
1.4　EDの終了　277

第2章　EDのコマンドの解説　278

2.1　カーソルの移動　278
2.2　挿　入　280
2.3　削　除　281
2.4　検　索　281
2.5　置　換　282
2.6　ファイルの読み込み・書き出し　285
2.7　カットバッファ　286
2.8　その他の編集コマンド　287
2.9　ファイルのセーブ・編集の終了　289
2.10　複数ファイルの編集　290

第3章　EDのスイッチ付き起動　292

第4章　EDの操作例　294

第5章　EDのコマンド一覧　296

付　録

付録 1　ASCII制御コード　301
付録 2　エスケープシーケンス　303
付録 3　テレビコントロールコード一覧　307

索　引　308

コマンド別目次

ATTRIB	ファイルの属性の設定・解除	48
BACKUP	ハードディスクからフロッピーディスクに、1つまたはそれ以上のファイルのバックアップ作成	50
BIND	オーバーレイXファイルの作成・変更	52
BREAK	CTRL + C のチェック機能の設定	55
CACHE	CPUキャッシュのON、OFFを設定	56
CHDIR(CD)	カレントディレクトリの変更・表示	57
CHKDSK	ディスクの状態の報告	58
CLS	ディスプレイの消去	60
COMMAND	コマンドプロセッサCOMMAND.Xの起動	61
COPY	ファイルのコピー・連結	64
COPY2	ハードディスク上のファイルのフロッピーディスクへの退避・フロッピーディスク上の退避ファイルのハードディスクへの復帰	68
COPYALL	ファイルおよびディレクトリのコピー	72
CTTY	入出力デバイスの変更	74
DATE	日付の表示・設定	75
DEL(ERASE)	ファイルの削除	77
DIR	ディレクトリ（ファイル名一覧）の表示	79
DISKCOPY	ディスク全体のコピー・比較照合	82
DRIVE	ドライブの種類、状態の表示・ドライブ名の交換	88
DUMP	16進表現・文字によるファイル内容の表示	90
ECHO	バッチ処理中におけるコマンド行表示の設定・メッセージの表示	181
EXIT	子プロセスとして起動されたCOMMAND.Xからの親プロセスへのリターン	91
FASTIO	IO関連の高速化	92
FASTOPEN	ファイル名の管理環境設定	93
FASTSEEK	デバイスの管理領域サイズの設定	94
FC	ファイル内容の比較	95
FIND	ファイル中の指定文字列の検索	97
FOR	コマンドの反復実行	183
FORMAT	ディスクのフォーマット(初期化)	99
GOTO	バッチ処理の流れの変更	184
HIS	ヒストリ行列の表示	108
IF	条件設定によるバッチ処理	185
KEY	ファンクションキーなどの設定	110
MEMFREE	使用可能なメモリサイズの表示	114
MKDIR(MD)	新しいディレクトリの作成	115
MORE	1画面ごとの表示	116

MOVE　ファイルの移動……117
PATH　外部コマンド検索のための PATH の設定……118
PAUSE　バッチ処理の一時停止……187
PR　ファイル内容のページング……120
PROCESS　プロセスの情報を表示……122
PROMPT　プロンプトの設定……125
RECOVER　不良クラスタを含むファイルまたはディスクの修復……127
REM　バッチ処理におけるコメントの表示……187
REN(RENAME)　ファイル名の変更……130
RESTORE　BACKUP コマンドでバックアップされたファイルの復元……132
RMDIR (RD)　ディレクトリの削除……134
SCREEN　画面モードの設定……135
SET　環境文字列の値の設定……137
SHIFT　バッチ処理中におけるパラメータのシフト……188
SORT　データのソート(並べかえ)……139
SPEED　RS-232C インターフェイスに対するパラメータの設定・起動……140
SUBST　仮想ドライブの割り当ての設定・解除……144
SWITCH　メモリスイッチの設定……146
SYS　Human68k システムの転送……161
TEMP　テンポラリファイルのパスの設定……163
TIME　時刻の表示・設定……164
TIMER　時刻の表示・アラームのセット……165
TREE　ディレクトリ構造のツリー形式での表示……168
TYPE　ファイル内容の表示……169
USKCGM　外字の作成・登録・削除……170
VER　システムのバージョンの表示……175
VERIFY　ベリファイ機能の設定……176
VOL　ディスクのボリュームラベルの表示・変更……177
WHERE　ファイルの検索……179

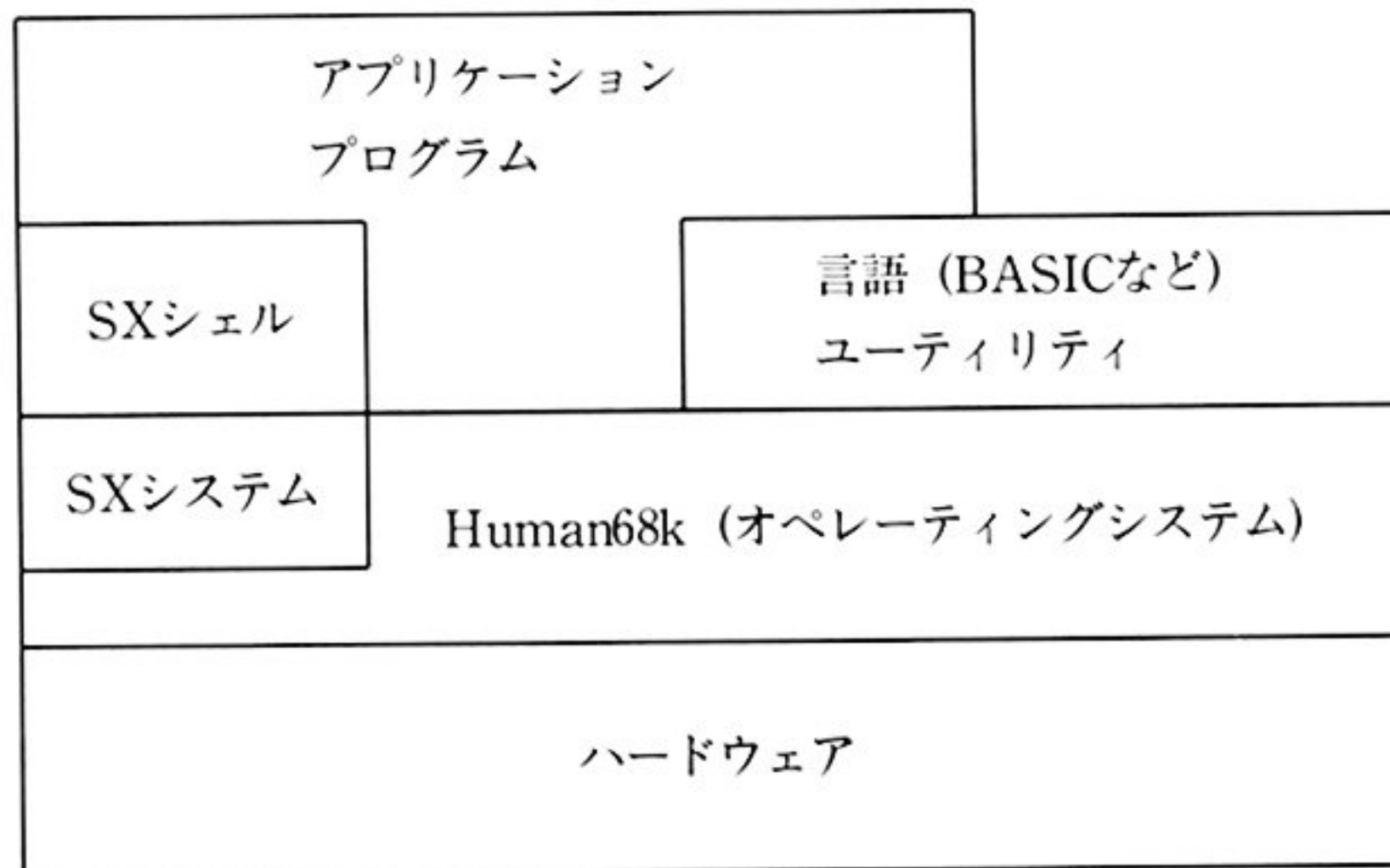
第1部

Human68k

第１章　Human68k の概要

1．1　Human68k とは

Human68k は、本機のハードウェアの持つ高度な性能を最大限に発揮するために、独自に開発されたオペレーティングシステム（OS）です。

OS とは、ハードウェアとアプリケーションプログラムの間に立って、全体の機能をコントロールするための基本ソフトウェアのことです。OS は、表立った活躍をするわけではありませんが、アプリケーションプログラムが個々の機能を最大限に発揮できるように陰で働く、いわば〝縁の下の力持ち〟的な存在である、といえます。たとえば、Hキーを押すと、「H」が入力されたことを確認し、ディスプレイに表示します。また、文字をプリンタから印刷したり、フロッピーディスクにデータを保存したりします。

コンピュータを全体として見た場合には、まずハードウェアがあり、その上にオペレーティングシステム（OS）があります。この OS（本機の場合は Human68k です）が、お使いのアプリケーションプログラムの実行環境を提供しているのです。

Human68kは、ハードウェアとアプリケーションプログラムの両方を管理します。付属のSX-WINDOWやX-BASICも、Human68kのコントロールの下で動作します。

１．２　Human68kの特長

１．２．１　バックグラウンド処理

Human68kは、パソコンを使ったネットワーク構築への環境が整っています。このネットワーク構築への対応については、大きく分けて次の２つがあります。その一つがこの節で取り上げる「バックグラウンド処理」、もう一つは次節で取り上げる「ネットワーク環境への対応」です。

バックグラウンド処理とは、ある処理を行いながらその裏（バックグラウンドプロセス）で別のプログラムを同時に実行するというもので、「並行処理」ともいいます。

たとえば、通常複数台のパソコンを使ってネットワークを組むと、そのうちの一台が「ファイルサーバー」として占有されてしまい、パソコンとしての機能が使えなくなってしまいます。ところが、バックグラウンド処理を使うと、ファイルサーバーとして使いながら、同時に通常の処理もできます。

また、複数台のパソコンを使ったネットワーク処理中の並行処理のほかに、一台のパソコンで複数の処理を並行して行うこともできます。たとえば、ワープロで文章を入力・編集しながら、ディスクに収められている文書をプリンタに出力したりできます。

このようなバックグラウンド処理は、ネットワーク処理用のボードやこの機能を組み込んだソフトウェアが登場してはじめて実際に行えるようになります。現状では、上記のようなバックグラウンド処理は行えませんが、Human68kでは、このバックグラウンド処理のサンプルとして、ウェイクアップタイマ機能(TIMERコマンド)を用意しています。TIMERコマンドでは、ある処理を行いながら指定した時刻にメッセージを表示する、ADPCMファイルを再生する、テレビをコントロールする、などができます。TIMERコマンドの詳細については、「第５章　コマンド TIMER」をご覧ください。

なお、バックグラウンド処理を行うには、あらかじめCONFIG.SYSに並行処理を行うプログラム数やレベル（実行間隔）、タイムスライス値（実行時間）を、「PROCESS=」を使って指定しておく必要があります。また、バックグラウンドプロセスを含めた各プロセスの状態などは、「PROCESSコマンド」を使って確認することができます。「PROCESS=」は第７章を、「PROCESS コマンド」は第５章を参照してください。

1.2.2　ネットワーク環境への対応

ネットワーク環境への対応とは、今後実現されるネットワーク処理に備え、Human68kの機能として、「ファイルの共有化とロック」、そして「仮想ドライブへの対応」などを準備しているというものです。これらの機能はネットワークを構築するときになくてはならない機能です。

・ファイルの共有化　ネットワークを組んだときに複数のプロセスから、同じ一つのファイルをアクセスして使えるようにする機能です。このファイルの共有化によって、たとえば、ネットワークで結ばれた複数のユーザーが同じファイルを共有して使えるようになります。

・ファイルのロック　共有したファイルに対して書き込みを行うときなどに、ファイルの一部を保護（アクセス禁止）するものです。このようにファイルをロックしないと、アクセスしたファイルが自由に書き込まれてしまうという危険性が生じてしまいます。

・仮想ドライブ　ネットワークで結ばれたときに、他のパソコンのドライブをあたかも自分のドライブの中のディレクトリとして参照したり、逆に、自分のドライブの中のディレクトリにドライブ名をつけて、そのドライブ名を他のユーザーにも参照させたりすることです。つまり、実体としては存在しないドライブ（ディレクトリ）に、実在するディレクトリ（ドライブ）を割り当てて使うわけです。

なお、ファイルの共有とロックは、あらかじめCONFIG.SYSの「SHARE=」で〈共有ファイル数〉と〈ロック領域数〉を指定しておく必要があります。

また、仮想ドライブの設定は、SUBSTコマンドを使って行いますが、仮想ドライブのドライブ数は、CONFIG.SYSの「LASTDRIVE=」で指定しておきます。

1.2.3　ヒストリデバイスドライバ

Human68kは、ヒストリデバイスドライバHISTORY.XをCONFIG.SYSに組み込むこと（コマンド行からも実行可能）で、コマンドモードでのキー入力や編集が次のように効率的に行えます。

●コマンド行での編集機能

これまでのテンプレート機能に加え、テンプレートからコマンド行に入力した文字列を、カーソルキーやCTRLを使って簡単に編集できます。

●テンプレートの操作

テンプレート機能は、HISTORY.Xが組み込まれていなくても利用できますが、HISTORY.Xを組み込むことによってテンプレートの引き数だけをコピーすることもできます。

●入力文字列のヒストリの検索と再入力（ヒストリ機能）

これまでのHISコマンドを使ったヒストリ機能でも、入力済みのコマンドをさかのぼって呼び出すことができましたが、HISTORY.Xではこの機能をさらに拡張しました。

●ディレクトリ変更履歴の検索と再入力

ハードディスクを使っているときには、CHDIR (CD) コマンドを使ってディレクトリを頻繁に移動しますが、このディレクトリ変更だけのコマンドをヒストリ機能のように呼び出して実行できます。

●エイリアス（別名定義）機能

スイッチやオプションの多いコマンドを、そのつど入力しているのは煩わしいものです。このようなコマンドを短い名前で定義しておき、この名前を入力するだけで長いコマンドが簡単に入力できます。

●簡易バッチ機能

これまで入力したコマンドと同じコマンドをまた入力して実行させるとき、そのつど入力するのは大変です。ましてや連続したコマンドを順に実行するときなど、複数のコマンドを覚えておくのは、煩わしいことです。

HISTORY. Xでは、以前に実行したコマンドのリストを表示し、その中から実行したいものを選ぶだけで、簡単に実行できます。

ヒストリ機能の詳細については、「6．4　ヒストリデバイスドライバ」をご覧ください。

1．2．4　ファイルアクセス

Human68kでは、将来の大容量デバイス（光磁気ディスク、CD、大容量ハードディスク、DATなど）に対応し、大容量ファイルにもアクセスできるように、1クラスタの容量を1Kバイト、2Kバイト、4Kバイト・・・32Mバイトと変えられます。この結果、大容量デバイスを接続し、大容量ファイルを扱いたい場合にも、アクセス可能です。たとえば1クラスタを64Kバイトとすると4G（ギガ）バイトの大容量ファイルでもアクセスできるようになります。

1．2．5　対話型メニューコマンドの採用

誰でも簡単にしかも画面と対話しながらコマンドを操作できるよう、対話型メニューコマンドが用意されています。これにより、初心者でも比較的簡単にコマンド操作ができ、また、デモンストレーションなどで複数の人が利用するときにも便利です。

1．2．6　ハードディスクのパーティション（領域分割）

Human68kでは、ハードディスクの領域を分割して、ドライブを複数割り当てられるFormatコマンドを搭載しています。たとえば、82MBのハードディスクの場合。25MBをドライブA、残りの57MBをドライブBとすることができます。また、このように領域を分割したときにどのドライブから起動するかを、起動時にメニューを表示させて自動起動するドライブを選択できます。

1.3 Human68k ver.3.0の特長

1.3.1 CPU種類の判別

起動時、CPUの種類を判別して起動できるようになりました。

Human68k ver.3.0は、X68000シリーズ上でも起動することができます。そのため、CPUがMC68000なのか、MC68030なのかを判別して起動できる機能をつけました。

1.3.2 SCSIデバイスドライバの標準装備

FDから起動した場合などでも、デバイスドライバ（SCSIDRV.SYS）の登録なしに、SCSIドライバが登録できるようになりました。SCSIドライバをROM内に持っているので、起動時にSCSIDRV.SYSを読み込まなくてもSCSIデバイスを使用することができます。

1.3.3 CONFIGファイル機能の充実

・CONFIG.SYSを自由に変更して起動したり、色々なCONFIG.SYSを選択して起動したりできる機能を追加しました。
このサンプルプログラムとしてCONFIGED.Xを用意しています（詳細は「第7章 システムの構築 EXCONFIG」参照）。

・デバイスドライバを登録する前にCOMMAND.Xを使うことができるようになったため、カレントディレクトリを移動してから、そのディレクトリに設定された環境で起動できるようになりました。

・起動したあとでもドライブの管理情報を変更できるようになったため、フロッピーディスクの種類を自動判別することなどが可能になりました（詳細は「7-3 FDDEVICEドライバについて」参照）。

・IOCACHE管理、FAT管理、DIR管理の一部を取り替えることができるようになりました。

・ディレクトリのサーチ機能が強化され、ファイル検索や新規作成時に、直前に実行されたディレクトリ位置の続きから検索するように指定できるようになりました。
このためファイル検索をするときに、毎回はじめから検索する場合に比べ高速に検索することができるようになりました。

1.3.4 MIDI対応、OPMDRV3.Xの搭載

・FM音源／MIDI／ADPCMを任意のチャンネル（1～25）に割り当てて鳴らすことができます。

・ADPCM音を128種類まで登録することができます。

・トラック数は80で、25トラックまで同時演奏することができます。

・１トラックで和音を演奏することができます（MIDIのみ）。

・モジュレーション／ピッチベンドのコマンドは、そのままFM音源に対しても機能します。

・常にMIDIクロック（F８）を出力しているので、外部シーケンサーを同期演奏させることができます。
外部同期に設定すれば、外部シーケンサーでコントロールすることもできます。
任意の小節でのスタート／ストップができます。

・SOUND PRO-68KのSNDファイルで音色設定ができます。

・MMLデータの中にチャンネル変更コマンドを自由に設定できます。

・MUSIC PRO-68K[MIDI]のMUSファイルをそのまま演奏できます。

・OPMDRV3.X上でMUSIC PRO-68K、MUSIC PRO-68K[MIDI]がそのまま利用できます。

注意
Human68k　ver.3.0システムディスクに収められているコマンドおよびデバイスドライバは、Human68K　ver.3.0上でお使いください。

第2章 Human68kの起動

2.1 イントロダクション

本章では、Human68kの起動の方法、画面上に表示されるプロンプト、Human68kシステムディスクのバックアップコピーの作成方法などについて解説します。

Human68kを操作する際には、本章の解説に従って、まずHuman68kの入っているシステムディスクのバックアップコピーを作成してください。今後Human68kを実行するときには、このバックアップコピーを使用してください。ハードディスク内蔵タイプのX68030シリーズをご使用の方が、Human68kを操作する際には、本章の解説に従って、まず内蔵ハードディスクにシステム、辞書ディスクの内容を転送し、内蔵ハードディスクからシステム起動できるように設定しなおしてください。今後Human68kを実行するときは、内蔵ハードディスクからシステム起動して使用してください。

2.2 Human68kの起動

ここでは、本機の電源を入れてから Human68k のコマンドモードにするための手順を解説します(コマンドモードとは、システムが画面に"A>"、"B>"……などと表示して、コマンドの入力待ちをしている状態のことをいいます)。

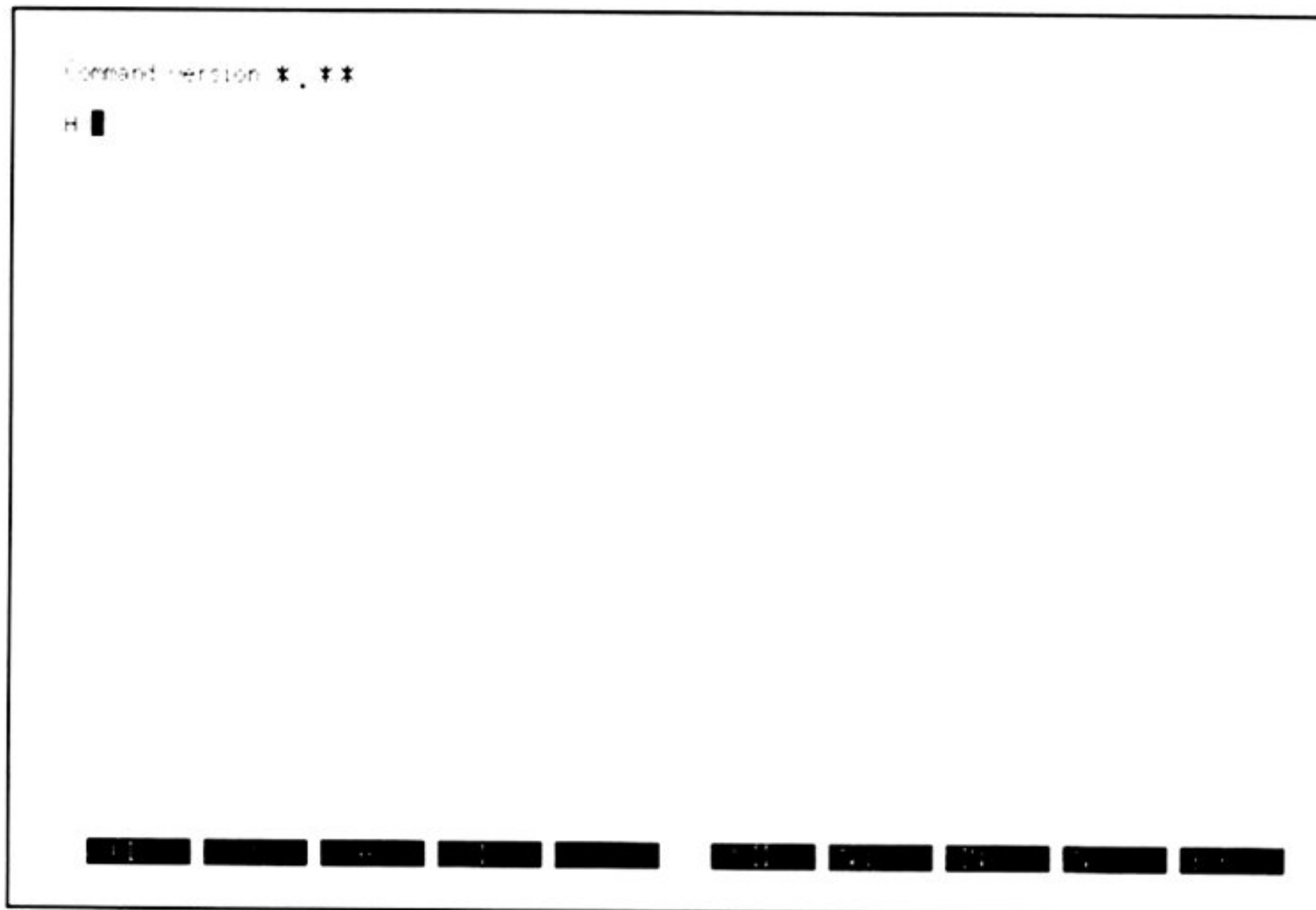
(1) まず、本機のメイン電源を"入"(ON)にしてください。次に、必要な周辺機器(専用カラーディスプレイテレビ、プリンタなど)が接続されていることを確認し、それらの電源を入れてください。続いて、本体前面にある電源スイッチを"入"(ON)にしてください。
各スイッチ、コネクタの位置などコンピュータの操作方法がよくわからないときは、「取扱説明書」を参照してください。

(2) ハードディスクを内蔵していないX68030シリーズをご使用の方は、バックアップされたシステムディスクをフロッピーディスクドライブ0(ドライブA)に挿入してシステム起動してください。ハードディスク内蔵タイプのX68030シリーズをご使用のかたは、内蔵ハードディスクからシステム起動してください。

なお、ハードディスク内蔵タイプで、フロッピーディスクドライブ0に挿入したシステムディスクやアプリケーションソフトから起動させるときには、[OPT. 1]を押しながらリセットスイッチを押し、[OPT. 1]を押したままドライブ0のアクセス表示用ランプが2回赤く点灯するのを確認してください。このあとドライブ0からシステムが起動されます([OPT.1]を使ったフッロピーディスクドライブからの起動は、X68030シリーズ用のソフトのみです)。

この一連の操作により、Human68kその他のプログラムはシステムディスク上から本体のメモリへ読み込まれ、システムが起動します。

SX-WINDOWのデスクトップ画面からコマンドモードにするには、Human68kシステムディスクにはいっている〝COMMAND.X〟という名前のアイコンを、マウスの左ボタンでダブルクリック（2度続けてクリック）してください。

これにより、COMMAND.Xがメモリに読み込まれてHuman68kのコマンドモードになり、次のような画面が表示されます。

ここで、

A>

は、Human68kがコマンドの入力待ち状態であることを示します。

この〝A>〟のことを〝プロンプト〟と呼びます。ここで、〝A〟はカレントドライブがドライブAであることを表示しています。

カレントドライブとは、Human68kが現在作業をしているドライブのことです。

カレントドライブを変更するときは、プロンプトのすぐあとに、変更したいドライブ名とコロンを入力し、続いて[↵]（リターンキー）を押します。以後、[↵]はリターンキーを押すことを意味します。

カレントドライブを〝A>〟から〝B>〟に変更したいときは、

B：[↵]

と入力します。すると画面では、

B>

と表示され、カレントドライブが変更されたことを示します。

2.3　システム終了の方法

Human68kのシステムを終了するには、特別な操作は必要ありません。ただし、アプリケーションプログラムなどはすべて終了し、画面にプロンプトが表示されていることを確認してください。このとき、他のアプリケーションプログラムからコマンドプロセッサを2次的に呼び出していることがありますので、念のために、

EXIT ↵

と入力してみてください。他のアプリケーションプログラムから呼ばれたコマンドプロセッサの場合は、元のアプリケーションプログラムに戻ります。

作業中のプログラムが何もなければ、そのまま本体前面の電源スイッチを切って終了してください。

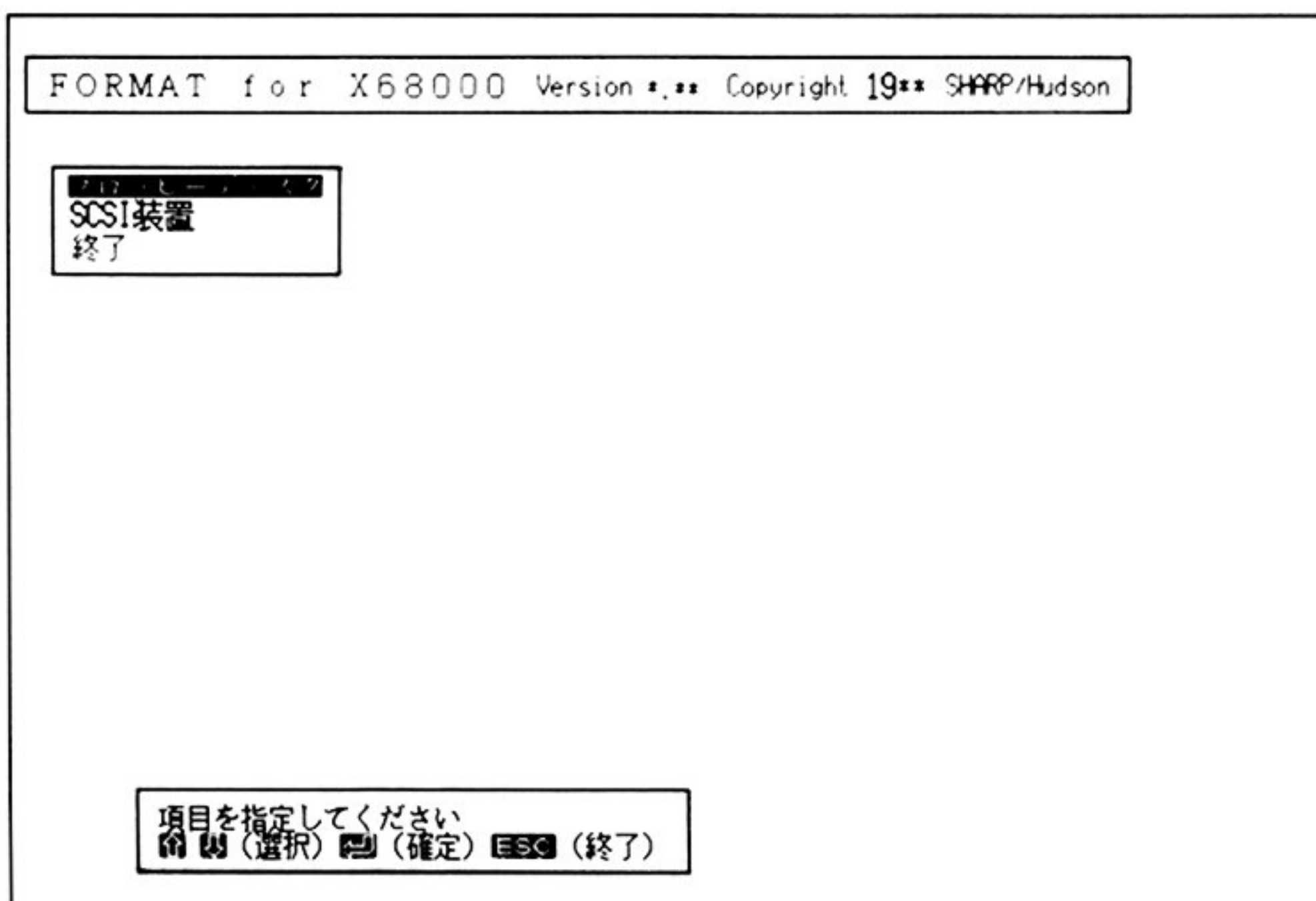
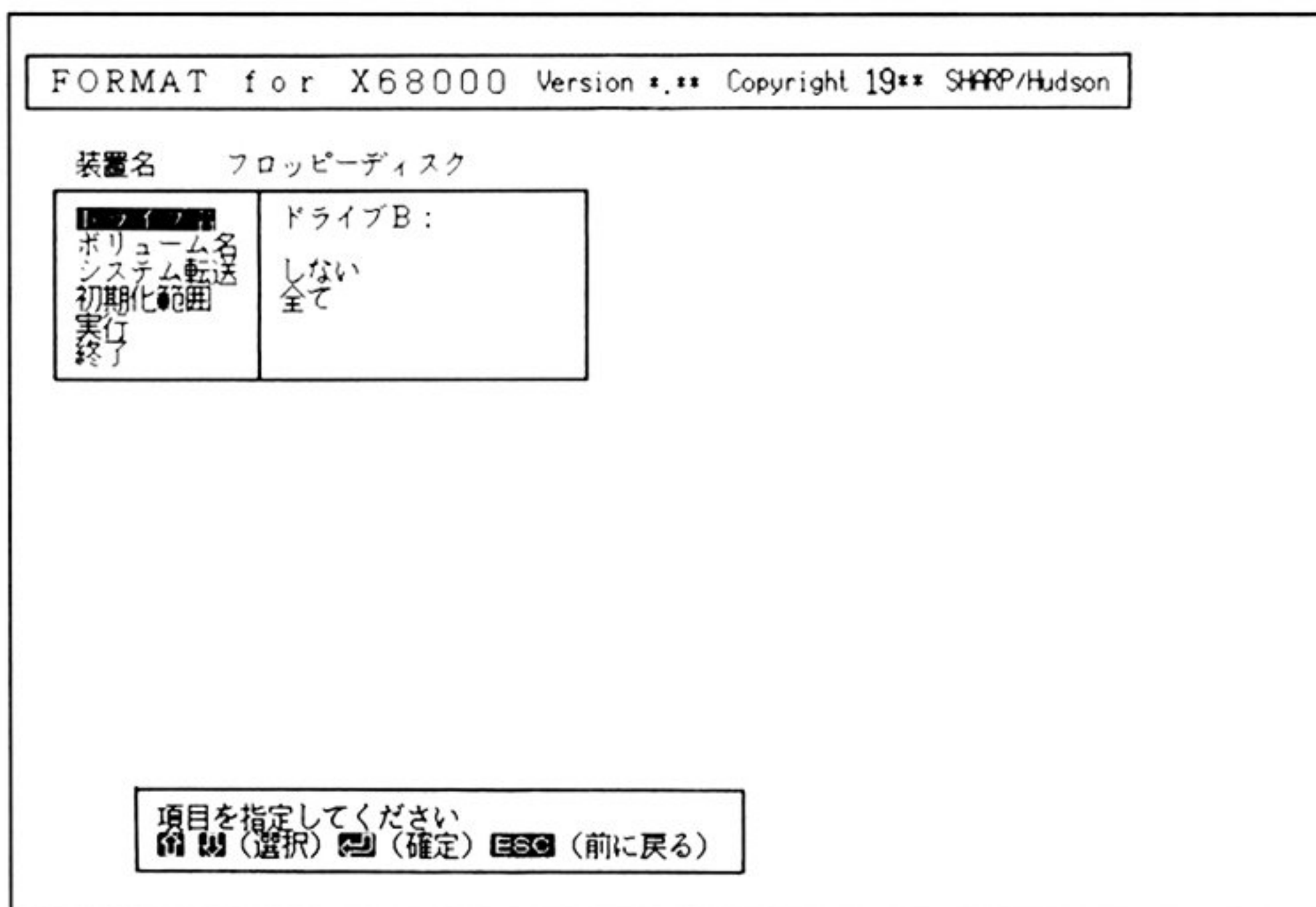
2.4　フロッピーディスクのフォーマット

新品のフロッピーディスクを使うときは、その前に必ず〝フォーマット〟をしなければなりません。フォーマットとは、Human68kがフロッピーディスクのデータを効率的に扱うために、〝セクタ〟とよばれる〝区画〟を決める処理のことです。紙と鉛筆による作業をする場合でも、白紙のノートをそのまま使うのではなく、自分が使いやすいように線を引いたりすることがあると思います。区画を決めることによって白紙の状態よりも情報が整理しやすくなるわけです。フォーマットもこれと同様です。

フロッピーディスクをフォーマットするときは、FORMATコマンドを使用します。(FORMATコマンドの詳細については第5章を参照してください)。

(1) コンピュータをシステム起動したあと、Human68kのコマンドモードに入り、プロンプトが表示されている状態で、次のように入力します。

FORMAT ↵

次のようなメニュー画面が表示されます。

(2) カーソルキーの ↑ ↓ で「フロッピーディスク」に合わせて、 ⏎ を押します。
画面には次のようなメニューが表示されます。

画面が次のようになっていることを確認してください。

ドライブ名　**ドライブB：**
（フォーマットするフロッピーディスクを挿入するドライブ名。）

ボリューム名
（フロッピーディスクを識別するためにディスクの中に記録する名前。必要に応じて付けますが、ここでは省略します。）

システム転送　**しない**
（Human68kのシステムを転送しません。転送するときは「する」に変えます。）

初期化範囲　**全て**
（フロッピーディスクのすべての領域を初期化します。一度初期化してあるフロッピーディスクのデータをすべて削除してフォーマットするときには、「管理領域のみ」とするとフォーマットが速くできます。）

もし、画面が上のような設定になっていないときは、↓ ↑ を使ってカーソルを設定し直す項目に合わせ、⏎ を押してください。メニューが表示されますので、↓ ↑ を使ってここで設定する内容にカーソルを合わせ、⏎ を押します。

(3) ドライブBに新しいフロッピーディスクを挿入します。
ドライブBとは、ハードディスクを内蔵していないタイプでは、ドライブ1
　　　　　　　　ハードディスクを内蔵しているタイプでは、　ドライブ0
を指します。

(4)　カーソルを「実行」に合わせ、[↵]を押します。
次のようなメッセージが表示されます。

ドライブB（2HDディスク）を初期化します
何かキーを押してください　[ESC]（前に戻る）

(5)　任意のキーを押します。
これで、ドライブBのフロッピーディスクへのフォーマットが始まります。
もし、ここで中止するときは、[ESC]を押してください。前の画面が表示されます。

フォーマットが終了すると、次のようなメッセージが表示されます。

初期化は終了しました　他のディスクを初期化しますか？
[Y]（継続）　[N]（前に戻る）　[ESC]（前に戻る）

(6)　終了するときは、[N]または[ESC]を押してください。
元のメニュー画面に戻ります。
もしここで、続けて別のフロッピーディスクをフォーマットしたいときは、[Y]を押してください。このあと、ドライブBに挿入されているフォーマット済みのフロッピーディスクを抜き、新品のフロッピーディスクを挿入して任意のキーを押してください。
(7)　カーソルを「終了」に合わせ、[↵]を押します。
FORMATコマンドが終了し、プロンプトが表示されます。

新品のフロッピーディスクをフォーマットするときはよいのですが、すでにデータが記憶されているフロッピーディスクをフォーマットすると、データはすべて消えてしまいます。フォーマットのことを〝初期化〟とも言いますが、これはデータがない枠だけが引いてある白紙の状態に戻すことから、そう呼ばれています。

したがって、必要なフロッピーディスクを誤ってフォーマットしてしまうことのないよう、FORMAT コマンドを使用するときは十分な注意が必要です。

2.5　フロッピーディスクのバックアップ

フロッピーディスクのバックアップは、大切なデータを誤った操作や予期せぬ事故から保護するために行います。フロッピーディスクにコーヒーをこぼしてしまった、必要なフロッピーディスクを誤ってフォーマットしてしまった、などのような場合、フロッピーディスクのバックアップコピーをとっておけばあわてる必要はありません。バックアップコピーを使用できるわけです。

Human68k のシステムディスクなど、大切なフロッピーディスクは作業の前に必ずバックアップコピーをとる習慣を身につけてください。通常の作業はこのバックアップコピーを使います。マスターのフロッピーディスクは万一に備えて保管しておきます。

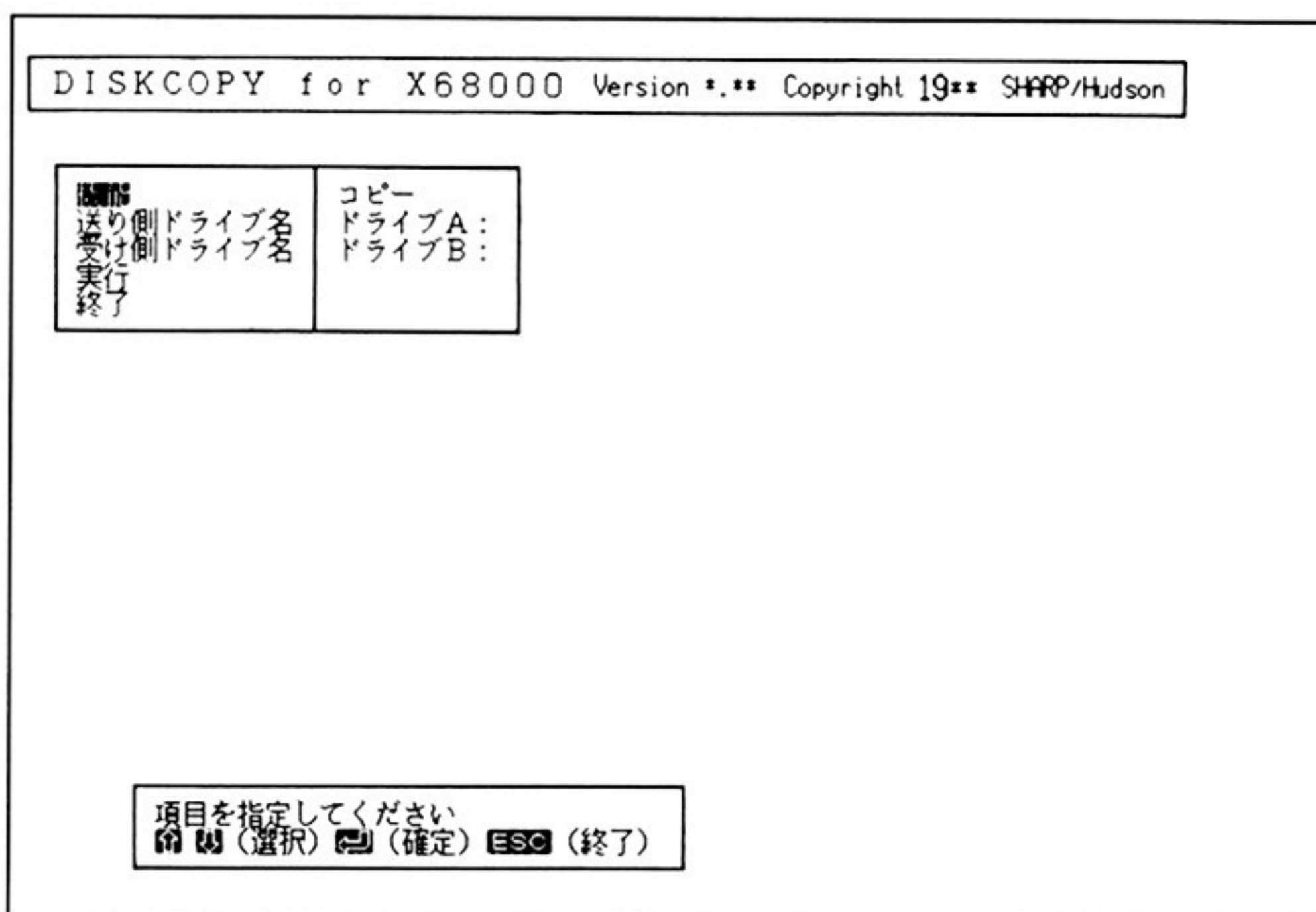
フロッピーディスクのバックアップコピーは以下の手順で行います。

(1) 新品のフロッピーディスクを FORMAT コマンドでフォーマットします。

(2) Human68k コマンドモードのプロンプトが表示されている状態で、

DISKCOPY ⏎

と入力します。

次のようなメニュー画面が表示されます。

画面が次のようになっていることを確認してください。

機能　**コピー**

(バックアップします。フロッピーディスクの内容を比較するときは、「比較」にします。)

送り側ドライブ名　**ドライブA：**

ハードディスク内蔵タイプは、ドライブB：

(コピーの原本になるフロッピーディスクを挿入するドライブ名。)

受け側ドライブ名　**ドライブB：**

ハードディスク内蔵タイプは、ドライブC：

(フォーマット済みのフロッピーディスクを挿入するドライブ名。)

もし、画面が上のような設定になっていないときは、↓↑を使ってカーソルを設定し直す項目に合わせ、⏎を押してください。メニューが表示されますので、↓↑を使ってここで設定する内容にカーソルを合わせ、⏎を押します。

(3) ドライブAに原本となるフロッピーディスクを挿入します。
ハードディスク内蔵タイプでは、ドライブBに挿入します。
ここで、原本となるフロッピーディスクを挿入するのは、ドライブ0となります。

(4) ドライブBにフォーマット済みのフロッピーディスクを挿入します。
ハードディスク内蔵タイプでは、ドライブCに挿入します。
ここで、原本となるフロッピーディスクを挿入するのは、ドライブ1となります。
フロッピーディスクのフォーマットは、「2．4 フロッピーディスクのフォーマット」の手順で作成してください。

(5) カーソルを「実行」に合わせ、[↵]を押します。
次のようなメッセージが表示されます。

ドライブAから、ドライブBへコピーします
何かキーを押してください　[ESC]（前に戻る）

もし、ここで中止するときは、[ESC]を押してください。前の画面が表示されます。
ハードディスク内蔵タイプでは、「ドライブBから、ドライブCにコピーします」と表示されます。もし、

容量の異なるディスクは、コピーできません
何かキーを押してください

と表示されたときは、任意のキーを押したあと、前の画面でドライブ名を確認の上、送り側ドライブ名をドライブBに、受け側ドライブ名をドライブCに変更してください。

(6) 任意のキーを押します。
これで、コピーが始まります。
コピーが終了すると、次のメッセージが表示されます。

コピーは終了しました　他のディスクをコピーしますか？
[Y]（継続）　[N]（前に戻る）　[ESC]（前に戻る）

(7) 終了するときは、[N]または[ESC]を押してください。
元のメニュー画面に戻ります。
もしここで、続けて別のフロッピーディスクにコピーしたいときは、[Y]を押してください。このあと、ドライブB（ハードディスク内蔵タイプではドライブC）に挿入されているコピー済みのフロッピーディスクを抜き、別のフォーマット済みのフロッピーディスクを挿入して任意のキーを押してください。

(8) カーソルを「終了」に合わせ、↵を押します。
DISKCOPYコマンドが終了し、プロンプトが表示されます。

２．６　自動実行バッチファイル(AUTOEXEC．BAT)について

市販のアプリケーションプログラムなどの入ったフロッピーディスクをセットして本機をリセットしたとき、Human68k の起動メッセージが出力されたあとに、自動的にそのアプリケーションプログラムの実行が開始されることがあります。これは、そのフロッピーディスク上に〝AUTOEXEC. BAT〟と呼ばれる特別なファイルが収められていて、これにより自動的にそのアプリケーションプログラムが実行されるようになっているからです。

この〝AUTOEXEC. BAT〟ファイルは、「自動実行バッチファイル」と呼ばれるもので、後述するバッチファイルの一種です。このファイルは自分で作ることもでき、システムの起動時に実行したいプログラムを立ち上げたり、いろいろな環境を設定したりする(RAMディスクに必要なファイルをコピーしたり、カレントドライブを B や C に設定したり……)のに利用することができます。

この自動実行バッチファイルの詳しい作り方などについては、「４．６バッチ処理」の項をご覧ください。

２．７　ファイル

ファイルは、コンピュータが管理するデータの集まり、と言うことができます。従って、単にテキストデータの集まりだけではなく、キーボートやプリンタなどの周辺機器もファイルと見なすことができます。第３章の「３．４予約ファイル名」で周辺機器を表すものが予約ファイル名となっていますが、これは、この概念によるものです。

Human68k では、データ、ファイル、ディレクトリの３者を、互いに関連づけてとらえると理解しやすいでしょう。情報の基本単位がデータで、データにあるまとまりを持たせたものがファイル、ファイルの集まりがディレクトリです。

Human68k を起動し、プロンプトの出ている状態で、

DIR ↵

と入力します。すると、ディスク上のファイルおよびディレクトリの一覧が、その大きさや作成日時とともに表示されます。ディレクトリには、〈dir〉という表示がついています。

これらは、本棚の扉を開き、そこに並んでいる本やバインダーの背の一覧表をながめることに相当します。１つ１つの本棚は、あるまとまった内容を持ち、それぞれ別の名前がつけられて並んでいます。また、数冊の本がまとめられ、１つのバインダーに入っていることもあります。このバインダー(「取扱説明書」ではフォルダーともいう)が（サブ）ディレクトリに相当するものです。

ファイルに含まれているデータを見るには、

TYPE　〈ファイル名〉 ↵

と入力します。画面には指定したファイルのデータ(内容)が表示されます。たとえば、

TYPE　CONFIG. SYS ⏎

とすると、次のように表示されます。

```
FILES    =15
BUFFERS  =20 1024
     ⋮
```

DIR、TYPE コマンドの詳細は、第5章を参照してください。

2．8　内蔵ハードディスクの初期化と領域確保

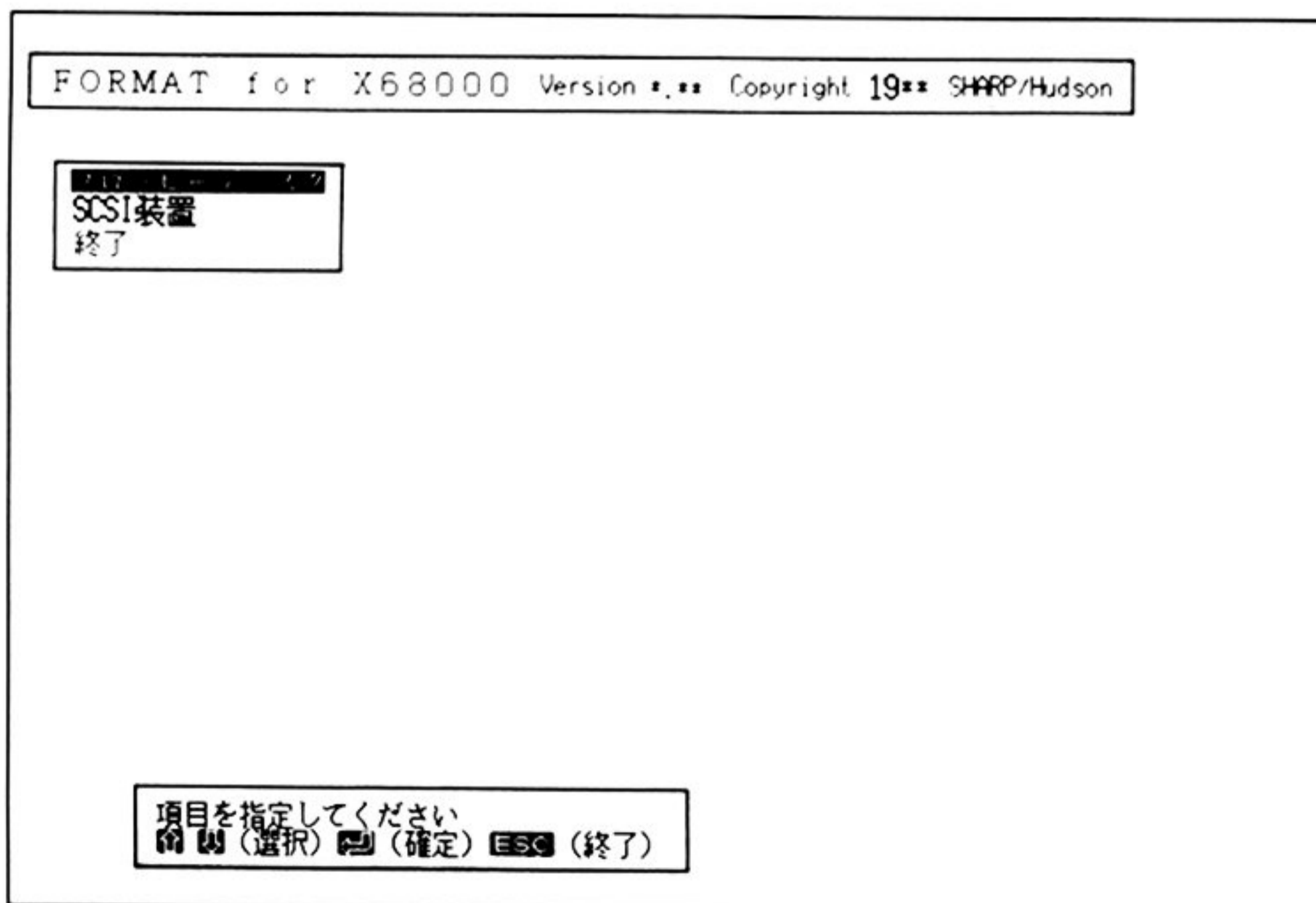
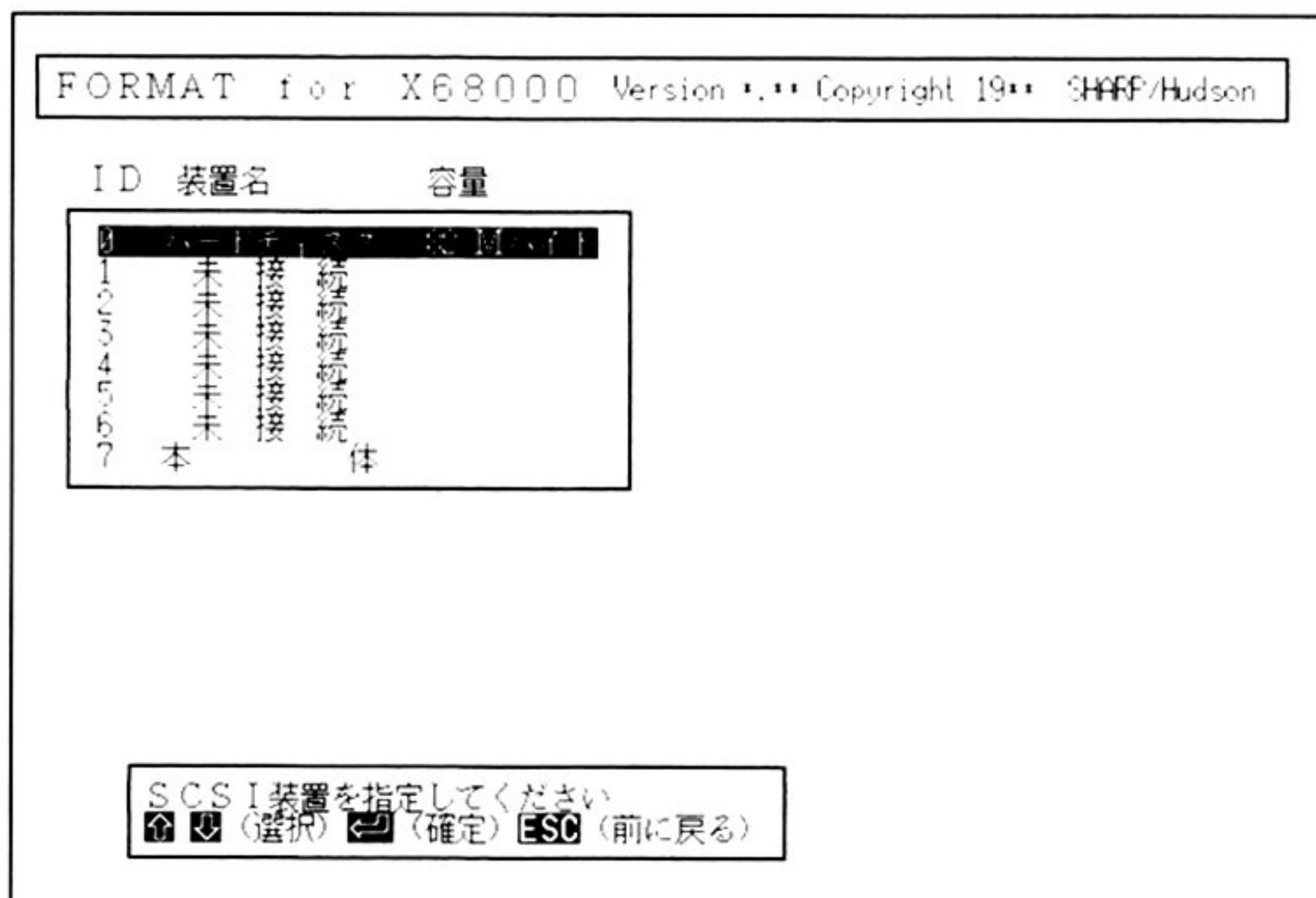
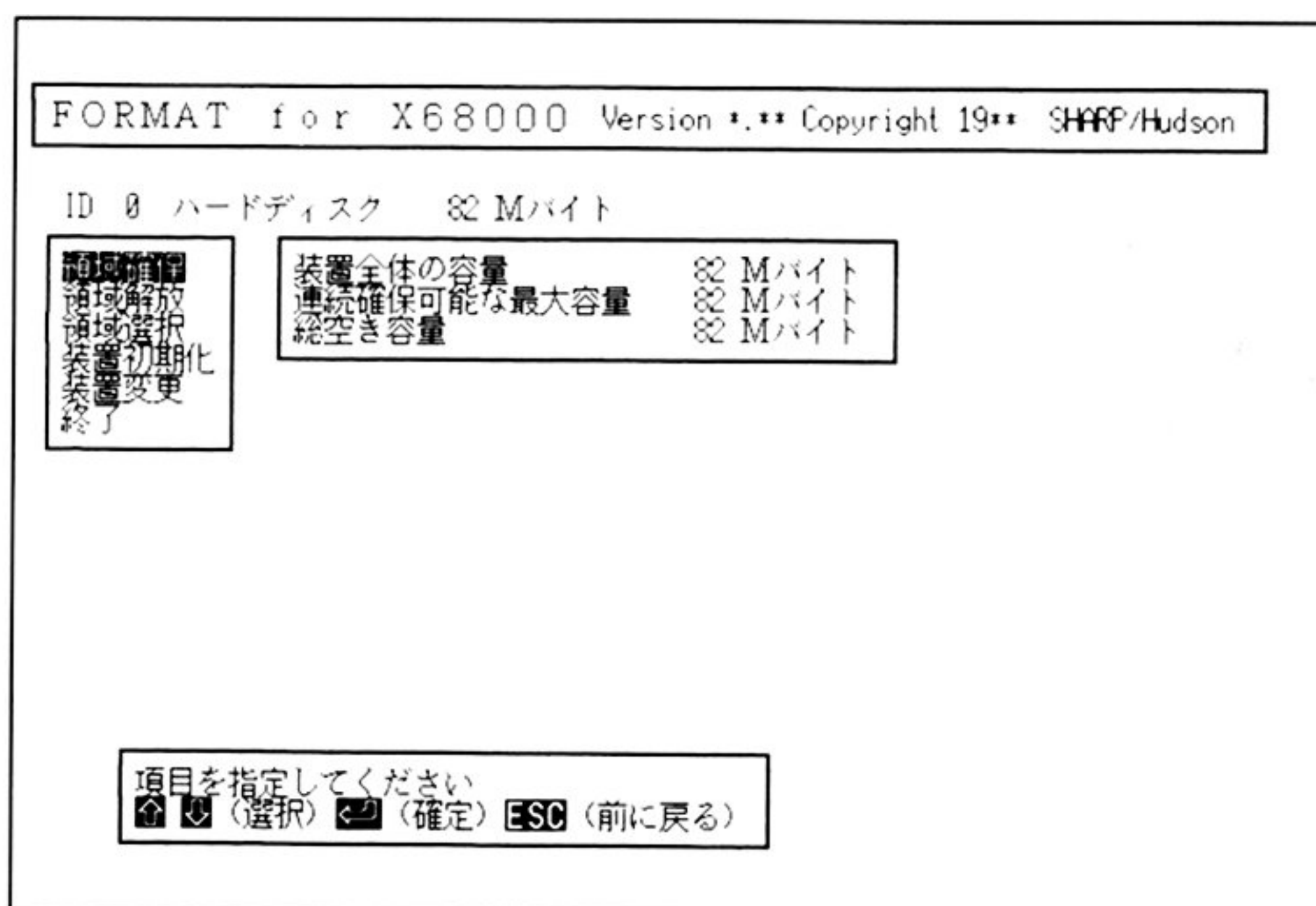
X68030シリーズで、本体に内蔵するタイプのハードディスクは、お買い上げになったばかりの状態(工場出荷時の状態)では、そのまま使用することができません。

標準でハードディスクが内蔵されているタイプや、増設用のハードディスクユニットを内蔵された方は、まず始めに、次の手順にしたがって、内蔵ハードディスクの装置の初期化、並びに領域確保を行うと、内蔵ハードディスクを使用することができます。

以降の説明では、本体外部にフロッピーディスクドライブを接続されていないものとして説明を進めます。

(1) コンピュータをシステム起動したあと、Human68kのコマンドモードに入り、プロンプトが表示されている状態で、次のように入力します。

FORMAT ⏎

すると、次のようなメニュー画面が表示されます。

(2) カーソルキーの ↑ ↓ で「SCSI装置」に合わせて、↵ を押します。
画面には次のようなメニューが表示されます。

(3) カーソルキーの ↑ ↓ でフォーマットするSCSI装置に合わせて（ここでは内蔵ハードディスクのID番号が0番なので、0を選択して）、↵ を押します。
画面には次のようなメニューが表示されます。

(4)　カーソルキーの ↑ ↓ で「装置初期化」に合わせて、↵ を押します。
画面には次のようなメッセージが表示されます。

装置全体を初期化します　よろしいですか？
Y （実行） N （前に戻る） ESC （前に戻る）

ここで Y を押すと初期化が開始されます。初期化が終了すると、元のメニュー画面に戻ります。

(5)　カーソルキーの ↑ ↓ で「領域確保」に合わせて、↵ を押します。
画面には次のようなメニューが表示されます。

容量では、3Mバイト以上を指定し、システム転送が"する"になっていることを確認してから「実行」を選択して↵を押すと、画面には次のようなメッセージが表示されます。

領域の確保を行います　よろしいですか？
Y （実行） N （前に戻る） ESC （前に戻る）

ここで Y を押すと領域確保が開始されます。
領域確保が終了すると、元のメニュー画面に戻ります。
ここで「終了」にカーソルを移動して↵を押すと、次のようなメッセージが表示されます。

ハードディスクの使用には再起動が必要です　リセットしますか？
Y （実行） N （終了）

ここで Y を押すと本体がリセットされ、再起動されます。

2．9　システム、辞書ディスクの内容を内蔵ハードディスクに転送する（内蔵ハードディスクからの起動）

システム、辞書ディスクの内容を内蔵ハードディスクに転送して、内蔵ハードディスクからシステムを起動するように設定すると、システム起動や日本語入力などが、フロッピーディスクに比べて高速に行え、快適な環境でお使いいただけます。

(1)　コンピュータが再起動したあと、プロンプトが表示されている状態で、次のように入力します。

COPYALL　/S　A：¥*.*　C：↵

(2)　終了したら、ドライブ１に辞書ディスクを挿入し、次のように入力します。

COPYALL　B：¥*.*　C：↵

以上で、前項のハードディスクのフォーマット画面の"領域選択"で、確保した領域を自動起動に設定することにより、内蔵ハードディスクからシステムを起動することができます。

また、内蔵ハードディスクからシステムを起動した場合、CONFIG.SYSファイルを書き換えることにより、辞書が使用できるようになります。

CONFIG.SYSファイルの書き換えは、次の手順で行います。

(1)　プロンプトが表示されている状態で、次のように入力します。

ED　CONFIG.SYS↵

すると、CONFIG.SYSファイルを読み込んでスクリーンエディタEDが起動し、次のように表示されます。

```
FILES     = 15
BUFFERS   = 20 1024
LASTDRIVE = Z:
KEY       = ¥KEY.SYS
USKCG     = ¥USKCG.SYS
BELL      = ¥BEEP.SYS
DEVICE    = ¥SYS¥PRNDRV.SYS
DEVICE    = ¥SYS¥ASK68K.SYS /DB:¥X68K.DIC /E¥ASK¥ENV1.ASK
DEVICE    = ¥SYS¥RSDRV.SYS
DEVICE    = ¥SYS¥OPMDRV3.X
DEVICE    = ¥SYS¥FLOAT2.X
DEVICE    = ¥SYS¥HISTORY.X /D¥HIS¥ /SH2,8,4
DEVICE    = ¥SYS¥IOCS.X
ENVSET    = 512 ¥STARTUP.ENV
```

※機種、バージョンによって、CONFIG.SYSファイルの内容が異なる場合があります。

(2)　スクリーンエディタ内の

DEVICE=¥SYS¥ASK68K.SYS /DB：¥X68K.DIC・・・

の行の”B：”の部分を”A：”に書き換えます。

```
FILES     = 15
BUFFERS   = 20 1024
LASTDRIVE = Z:
KEY       = ¥KEY.SYS
USKCG     = ¥USKCG.SYS
BELL      = ¥BEEP.SYS
DEVICE    = ¥SYS¥PRNDRV.SYS
DEVICE    = ¥SYS¥ASK68K.SYS /DA:¥X68K.DIC /E¥ASK¥ENV1.ASK
DEVICE    = ¥SYS¥RSDRV.SYS
DEVICE    = ¥SYS¥OPMDRV3.X
DEVICE    = ¥SYS¥FLOAT2.X
DEVICE    = ¥SYS¥HISTORY.X /D¥HIS¥ /SH2,8,4
DEVICE    = ¥SYS¥IOCS.X
ENVSET    = 512 ¥STARTUP.ENV
```

※ハードディスクの領域を複数確保し、領域1以外に辞書を転送する場合は、”B：”の部分を転送したドライブ名に書き換えてください。

(3)　次に ESC ・ E と続けてキーを押し、変更したファイルを保存してスクリーンエディタを終了します。

これで、システム、辞書ディスクの全ファイルが内蔵ハードディスクへ転送され、フロッピーディスクを入れなくても、システムが起動されX68030を快適に使用することができます。

※スクリーンエディタEDの詳しい使い方については、本書第2部を参照してください。

第3章　ファイルとディレクトリ

3．1　イントロダクション

第2章では、Human68k の基本的な操作、概念について解説しました。

本章では、ファイルを実際に扱う方法について、ファイル名のつけ方とその制限事項、ファイルのコピー、ワイルドカード、ディレクトリなどについて解説します。

3．2　ファイル名の付け方

ファイルは、処理を行うときにその対象となるデータのまとまりを指し示すものですから、そのファイルの内容が何であるかがわかるような名前を付けると便利です。

一般に、ファイル名は次のような形をしています。

ここでは、MASTERFL の部分だけを主ファイル名、ピリオドの後の".TXT"を拡張子と呼びます。

主ファイル名の長さは、半角で18文字、全角で9文字までです。また、拡張子としては、半角で3文字、全角で1文字まで使うことができます。

ただし、実際に Human68k が識別できる長さは、主ファイル名の先頭から8文字（全角だけなら4文字）分と、拡張子3文字（全角だけなら1文字）分です。

したがって、次の2つのファイル、

MASTERFL 1
MASTERFL 2

は同じファイル名と見なされることになります。また、

MASTERFL 1．TXT
MASTERFL 1．DOC
MASTERFL 1．REP

は、いずれも区別されることになります。

ピリオド以下3文字の拡張子は、そのファイルの種別を表すもので、Human68k では、以下の拡張子が特別な意味を持っています。

R　　**リロケータブル実行ファイル**
　　(メモリ空間上のどこにロードされても実行可能な、メモリイメージのファイル)
Z　　**アドレス固定実行ファイル**
　　(メモリ空間上の決められたアドレス上にロードされると実行可能となるファイル)
X　　**ソフトリロケータブル実行ファイル**
　　(メモリ空間上の任意のアドレス上に再配置された後、実行可能となるファイル)
S　　**アセンブリ言語ソースファイル**
O　　**オブジェクトファイル**
C　　**C言語ソースファイル**
A　　**ライブラリファイル**
SYS　**システムファイル**
BAT　**バッチ処理ファイル**
BAS　**X-BASICソースファイル**

この他、拡張子は〝.TXT〃、〝.DOC〃など、自由につけることができますが、アルファベット1文字だけの拡張子はシステム予約されていますので、使わないようにしてください。
ファイル名に使用できる文字は次のとおりです。

A〜Z　0〜9　&　#　(　)　@
_　^　{　}　!　カナ文字　全角文字

なお、小文字のa〜zも、ファイル名として使用することができます。ただし、小文字のa〜zについては、大文字のA〜Zと区別されてディスクに登録されます(なお、コマンドなどでこれを利用するときは区別されません)。

・ファイル名の例

MEIBO　　　MEMO
WKFL.100　　MSTFL.100
目次案.AAA　　目次案.BBB
計測.大　　計測.中　　計測.小

3.3　ワイルドカード

Human68kでは、ワイルドカードと呼ばれる2つの特殊な文字を使うことにより、ファイル名を扱う際に、柔軟性をもたせることができます。
これには、〝*〃と〝?〃があります。*(アスタリスク)は任意の文字列を表し、?(クエスチョンマーク)は任意の半角1文字を表します。
たとえば、

SAMPLE.001
SAMPLE.002
SAMPLE.003

というファイルをすでに作成しているとします。ディレクトリ中に多くのファイルがあるときに、このように主ファイル名の部分にSAMPLEという名前を持つファイルのみを表示したいときは、

```
DIR　SAMPLE.＊ ↵
```

または

```
DIR　SAMPLE.??? ↵
```

とします。また、

```
DIR　??????.＊ ↵
```

とすると、主ファイル名が6文字以下の名前のファイルをすべて表示します。

ワイルドカードは、このように便利なものですが、RENコマンドやDELコマンドを使用するときは十分に注意してください。

たとえば、

```
DEL　＊.＊ ↵
```

とすると、そのディレクトリのすべてのファイルが削除されてしまいます。

3.4　予約ファイル名

Human68kでは、システムで使用するために一般のファイル名として使えない名前があります。これを予約ファイル名といいます。予約ファイル名には次のものがあります。

AUX	補助入出力(RS-232C)機器との入出力を指定するときに使用
CLOCK	Human68k内部で使用されるファイル名。ユーザーは使用不可
CON	キーボードからの入力、またはスクリーンへの出力を指定するときに使用
LPT	プリンタへの出力を指定するときに使用 (漢字IN/漢字OUTコードを出力しない。ビットイメージ出力などに利用)
MIDI MIDIA MIDIAE MIDIB MIDIBE	MIDIデバイスへの出力を指定するときに使用
NUL	コマンドが入出力のファイル名を必要としているが、特にファイルを作成しないときに使用
OPM	FM音源デバイスへの出力を指定するときに使用
PCM	ADPCMデバイスとの入出力を指定するときに使用
PRN	プリンタへの出力を指定するときに使用(漢字IN/漢字OUTコードを出力する)

これらの名前は、それぞれ入出力機器を意味しています。なお、主ファイル名にこれらの予約ファイル名を用いたファイル、たとえば"PRN.ABC"などという名前のファイルを作成することはできません。

上記以外のファイル名であっても、英文字3文字だけで構成されるファイル名は、今後システムでの使用が考えられるため、一般には使わないようにしてください。

3.5　ファイルの取り扱い

本節では、ファイルをコピーしたり、作ったり、また消したりするにはどうしたらよいのか、具体的な例をあげて解説します。

3.5.1　ファイルのコピー

ファイル単位でコピーを行うときは、COPYコマンドを使用します。ファイルのコピーは、同一のディスク上でも2つのディスク間でも行うことができます。

ファイルのコピーは、以下の書式で行います。ここで、〈d1:〉〈ファイル名1〉はコピー元のドライブ名とファイル名、〈d2:〉〈ファイル名2〉はコピー先のドライブ名とファイル名を、それぞれ表します。

```
COPY　〈d1:〉〈ファイル名1〉　〔〈d2:〉〕〔〈ファイル名2〉〕
```

●同一ディスク上でファイルをコピーするとき

コピー元のファイル名とコピー先のファイル名は同じであってはいけません。

たとえば、

```
COPY　A:住所.001　A:住所.002
```

とすると、ドライブAのディスクにある"住所.001"がおなじディスクに"住所.002"という名前で作成されます。

● 2つのディスク間でファイルをコピーするとき

別のディスクにコピーするわけですから同じファイル名を指定することができます。

たとえば、

```
COPY　A:MSTFL.TXT　B:↵
```

とすると、ドライブAのディスクにある"MSTFL.TXT"がドライブBのディスクに、同じファイル名でコピーされます。

COPYコマンドは、このほかにもいろいろなコピーを行うことができます。詳しくは、第5章のCOPYコマンドの項をご覧ください。

3.5.2　ファイルの作成

ファイルを作るには、いろいろな方法があります。

1.スクリーンエディタやSX-WINDOWの日本語マルチフォントエディタを使って作る
　→テキストファイル

2.プログラムをコンパイル、アセンブルして作る→プログラムファイル

3.プログラム中などからデータを保存するために作る→データファイル

これ以外にも、まだいろいろな作り方があります。作るファイルの内容がさまざまならば、その作り方もさまざまなわけです。

しかし、どの場合も共通して言えることは、どんなファイルでも、「ある一定のデータのかたまり」に「規約に沿った名前」を付けたものである、ということです。

この項では、とりあえず簡単なテキストファイル(読むことのできる文字からなるファイルのこと)を前項で説明したCOPYコマンドを用いて作る方法について説明します。

次のように入力してください。このコマンドは、コピー元として指定したキーボードを表すCON(周辺機器を表すファイル名)からの入力を、コピー先として指定した"TEST.TXT"というファイルにコピーせよ、というものです。Human68kでは、周辺機器もファイルとしてとらえることができますのでこうした指定が可能となります。

COPY　CON　TEST.TXT ⏎

この行をタイプして、リターンキーを押すと、カーソルが2行下の先頭に移動したままとなります。ここで、何かテキスト（文字）を入力してください。

This is a test. ⏎
今日はよい天気です ⏎

そして最後に[CTRL]+[Z]([CTRL]キーを押しながら、[Z]キーを押す)を入力して、リターンキーを押します。[CTRL]+[Z]を入力することにより、CONの入力が終了したことをシステムに知らせるわけです。

^Z ⏎(^Zは、[CTRL]+[Z]を入力すると表示される記号)

これで、さきほど入力したテキストをその内容とするファイル"TEST.TXT"がディスク上に作成されます。試しに、このファイルを画面に表示してみます。

TYPE　TEST.TXT ⏎
This is a test.
今日はよい天気です

この方法は、スクリーンエディタやSX-WINDOWの日本語マルチフォントエディタのように自由に文字を編集できませんが、ほんの数行のテキストファイルを作ったりするにはたいへん簡単で便利なものです。第4章で説明するバッチファイルなどの作成にも利用することができますので、覚えておいてください。

3.5.3　ファイルの削除

いらなくなったファイルをディスク上から消すには、DELコマンドを用います。DELコマンドの書式は以下のとおりです。ここで、<d:>は削除したいファイルの入ったドライブ名を、<ファイル名>は削除したいファイルのファイル名を表します。

DEL　〔<d:>〕<ファイル名> ⏎

ここでは、前項で作成した"TEST.TXT"ファイルを削除してみましょう。

DEL　TEST.TXT ⏎

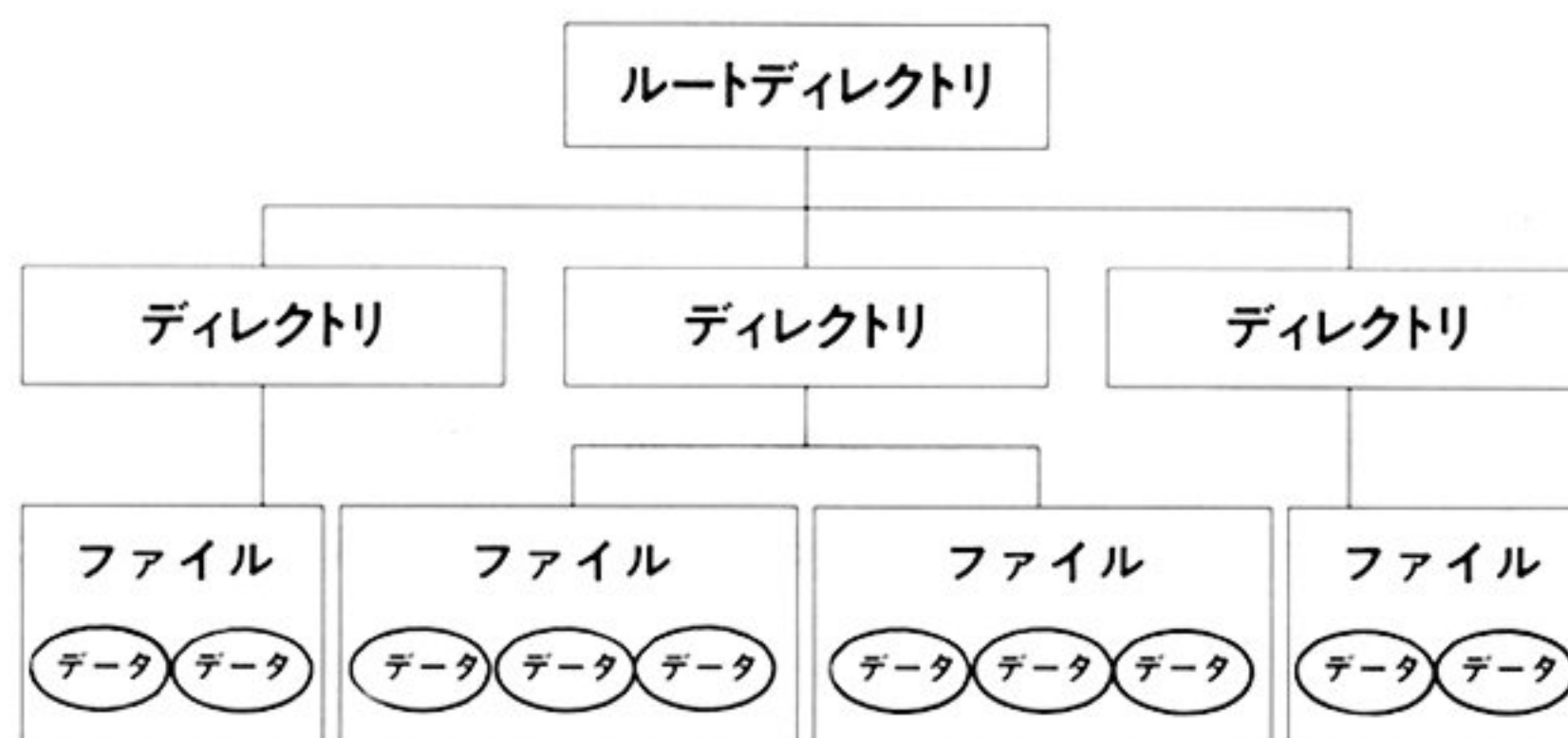
と入力します。ここでDIRコマンドを実行してみれば、ディスク上から、"TEST.TXT"ファイル

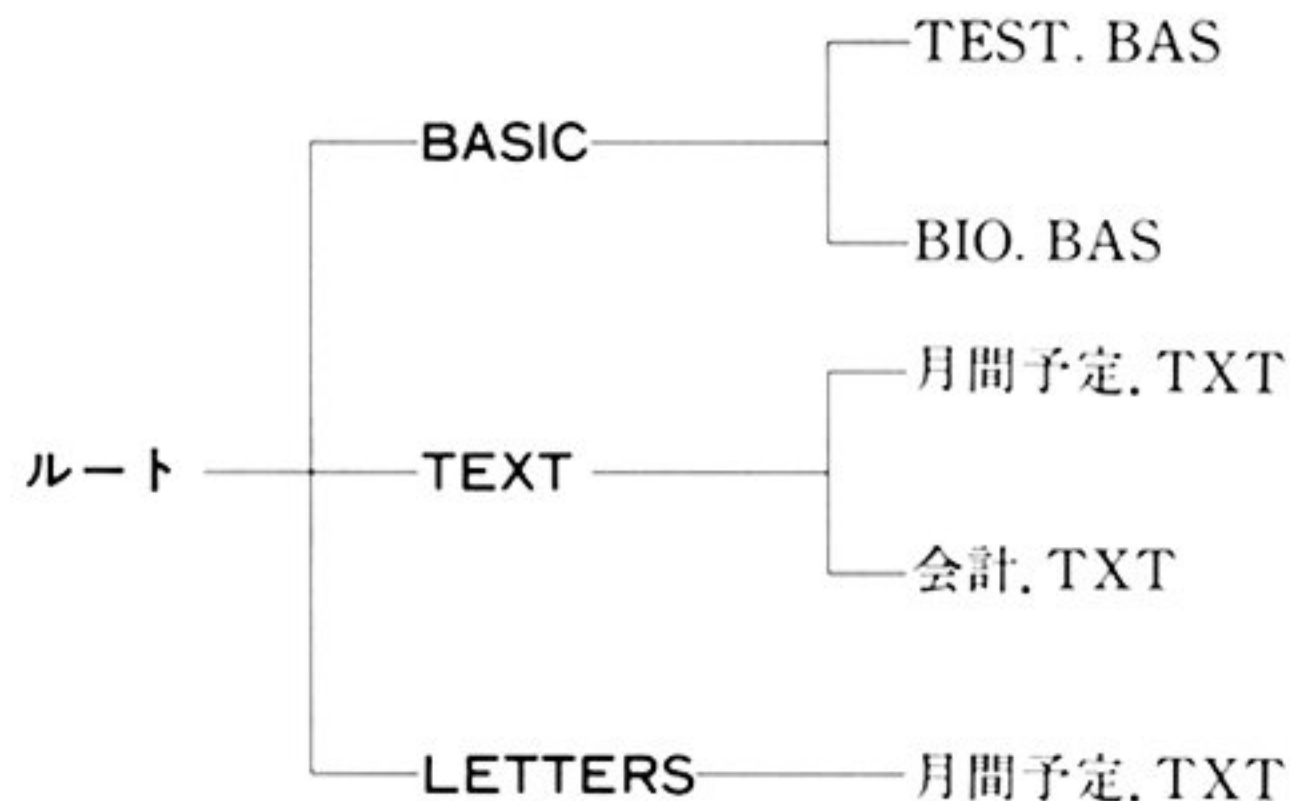
が消えてなくなっていることがわかります。

いったん消してしまったファイルは、もとどおりに戻すことはできません。削除する前に DIR コマンドや TYPE コマンドなどを使ってよく確かめるようにしてください。

DEL コマンドの詳細については、第5章の DEL コマンドの項をご覧ください。

3.6　ディレクトリ

ファイルとディレクトリの関係については、第2章で簡単に触れました。データの集まりがファイルで、ファイルの集まりがディレクトリでした。では、実際にはこれらの関係がどうなっているのか、次の図に示してみます。

この図は、階層ディレクトリ構造とか、またはたんに階層構造と呼ばれるものです。また、木構造（ツリー構造）などとも呼ばれます。なぜ木構造と呼ばれるかというと、ちょうど1本の木の構造にたとえることができるためです。ルートディレクトリの部分は、木でいうと根に当たります。ルートディレクトリの下にあるディレクトリ（サブディレクトリと呼ぶ）は木の幹に当たります。その下にあるディレクトリは枝に当たります。そのディレクトリの下のファイルは葉に当たります。実際には、ディレクトリはいく層でもとることができます。ルートディレクトリは根、ファイルは葉でその間にある幹や枝がディレクトリであると理解してください。根から葉に至る過程にはいくつものディレクトリが存在できます。このいく層ものディレクトリをたどる道筋を〝パス〟と呼びます。

あるディスク中に含まれているディレクトリおよびファイルが、次の図のような構成になっているものとします。

ルートディレクトリの下に〝BASIC〟、〝TEXT〟、〝LETTERS〟という3個のサブディレクトリがあり、さらにその下にそれぞれファイルがいくつか収納されています。

ここで、この例をよく見ると、〝月間予定 .TXT〟というファイルが、ディレクトリ〝TEXT〟とディレクトリ〝LETTERS〟の両方に入っていることがわかるでしょう。つまり、階層構造を用いることにより、同じ名前のファイルであっても、その属しているディレクトリが異なれば、互いに違うものとして区別することが可能となるわけです。このことは、ディスクを利用する際に、名前付けに対する自由度を大きくひろげてくれます。

ただ、これらの2つのファイルを特定するときには、いままで説明してきたような、主ファイル名と拡張子だけの組み合せでは表現できません。そこで登場するのが、さきに木にたとえて説明した、「パス」の概念です。すなわち、〝月間予定 .TXT〟を特定するには、

ルート→ TEXT →月間予定. TXT

であるか、または、

ルート→ LETTERS →月間予定. TXT

のどちらであるかを明示すればよいわけです。ここで、それぞれ〝ルート〟から〝月間予定 .TXT〟に至るまでの経路が、それぞれの「パス」となります。

同様のことが、他のファイルにもあてはまります。つまり、ディスク中におけるファイルの位置を特定しようとするときには、ルートディレクトリからそのファイルに至るまでの「パス」を明らかにすればよいわけです。たとえば、前例の〝TEST. BAS〟を特定するには、

ルート→ BASIC → TEST. BAS

とします。このように、あるファイルを特定することを、一般に「ファイルを指定する」とか、または単に「指定する」と呼びます。

3.7　ディレクトリの取り扱い

本節では、コマンドを使用して、階層構造中のディレクトリやファイルを操作する、具体的な方法について解説します。

3.7.1　パス名

前節では、ルートディレクトリとファイルの間の経路がパスであることを解説しました。正確には、「ある特定のディレクトリからある特定のファイルに至るまでの経路」がパスであり、それを文字で表したものがパス名です。具体的には、次のようになります。

これまでに解説してきたファイル名は、次のような書式をしていました。

〈ファイル名〉〔〈．拡張子〉〕

ここで、〈ファイル名〉は、半角文字で1～18文字の主ファイル名、〈拡張子〉は半角文字で1～3文字の拡張子です。角形カッコの中は省略可能な項目を表します。

もっと一般化して、これらの前に、ドライブ名として〈d：〉を指定することもできます。

〔〈d：〉〕〈ファイル名〉〔〈．拡張子〉〕

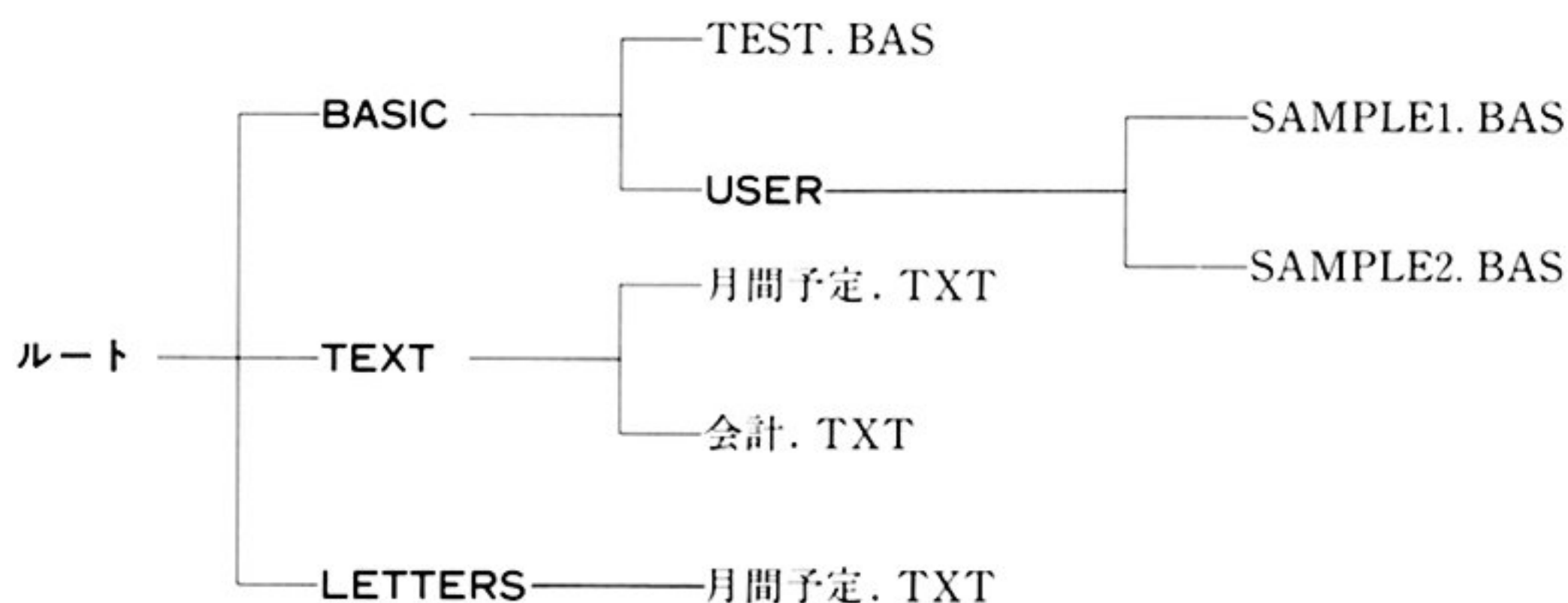
これが、階層ディレクトリでルートディレクトリとファイル名の間に何層かのディレクトリがパスとして作成されている場合は、次のような書式になります。

〔〈d：〉〕〔¥〕〔〈ディレクトリ名〉¥〕…〔〈ディレクトリ名〉¥〕〈ファイル名〉〔〈．拡張子〉〕

ファイル名までの、

〔¥〕〔〈ディレクトリ名〉¥〕…〔〈ディレクトリ名〉¥〕

が、パス名になります。ここで"¥"は、パス名を表すためにディレクトリ名およびファイル名を区切るための区切り記号です。ただし、いちばん最初の"¥"は、ルートディレクトリを表すシンボルとみなされます。

次のような階層構造があったとします（太字がディレクトリ）。

2つの〝月間予定.TXT〞を、それぞれパス名付きで指定すると、

¥TEXT¥月間予定.TXT
¥LETTERS¥月間予定.TXT

となります。
　ところで、このパス名付きのファイル指定ですが、これには、〝絶対〞指定と、〝相対〞指定の2種類の方法があります。
　絶対指定とは、ファイル指定のパス名がルート、すなわち〝¥〞で始まる指定方法のことを指します。ルートから全パスを指定することから、〝フルパス名による指定〞などとも言います。前の例は、2つともこの絶対指定によるファイル指定です。
　これに対して、相対指定とは、パス名が〝¥〞以外で始まる指定方法のことを指します。すなわち、カレントディレクトリ（次項で解説します。ここでは、たんにHuman68kが現在作業しているディレクトリのことである、と考えてください）から始めて、目的のファイルを指定する方法です。たとえば前の例で、カレントディレクトリが〝TEXT〞であるときに、〝会計.TXT〞を指定するには、

会計.TXT

とするだけでよいのです。この方法ですと、ファイル指定を行う際に、そのつどフルパス名を指定する手間が省けるため、階層構造の深いところにカレントディレクトリがあるときなどにも、簡単にファイル指定を行うことができるという利点があります。ただし、カレントディレクトリが明らかでない限り、指定されたファイルの位置を特定できない欠点があります。
　その点、絶対指定を用いれば、カレントディレクトリがどこであろうと、いつでもファイル位置の特定ができるため、正確な指定を行うことができます。
　この2つの指定方法は、どちらでも便利な方を選んでお使いください。
　ディレクトリの関係を親、子と表現することがあります。たとえば、前の例では、ルートディレクトリはサブディレクトリ〝BASIC〞、〝TEXT〞、〝LETTERS〞全部の〝親〞であり、逆にこれらサブディレクトリはルートディレクトリの〝子〞となります。
　Human68kでは、この〝親〞を、ピリオド2つ（..）で表すことができます。これは、ファイルの相対指定の中でも使うことができます。たとえば、カレントディレクトリが〝USER〞であるときに、〝TEST. BAS〞を指定するには、

..¥TEST. BAS

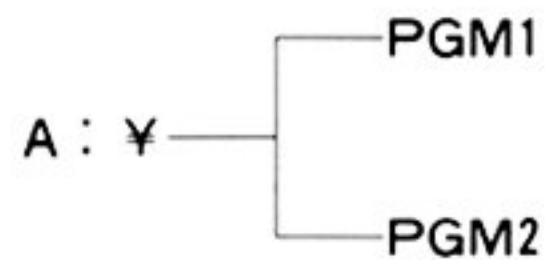
とすることができるのです。この方法は、階層構造が浅い場合はかえって指定がめんどうになりかねませんが、深い階層構造中で、「となりのディレクトリ」中のファイルを指定したりする場合は、大変便利に使うことができます。

3.7.2　ディレクトリの移動

階層構造では、〝カレントディレクトリ〟という概念があります。これは、ファイルに関する作業を行うにあたり、ある特定のディレクトリを中心にして操作ができるよう、システムが注目しているディレクトリのことを指します。これを操作する側から見て、〝いま自分がいるディレクトリ〟ととらえることもできます。

ディレクトリを移動することは、このカレントディレクトリを変更するということであり、階層構造中の自分の〝いる〟相対的な位置を変更するということを意味します。

ディレクトリを移動するにはCHDIRコマンドを使用します。

ドライブA中のディスクが、次のような階層構造になっているとします。

ディレクトリを、¥PGM 1から¥PGM 2に移動するときは、

CHDIR　¥PGM2 ⏎

と入力します。ディレクトリが移動したことを確認するときは、何もパラメータを付けずに、

CHDIR ⏎

と入力します。すると、

A：¥PGM2

と表示されます。これで、¥PGM2がカレントディレクトリになったことがわかります。

なお、ディレクトリの移動には、絶対指定のパス名のほかに、相対指定のパス名も用いることができます。とくに、〝親〟ディレクトリに変更する（戻る）ときには、

CHDIR　..

を使うと非常に便利です。

3.7.3　ディレクトリの作成

あるディレクトリの下の階層にディレクトリを作成するときは、MKDIRコマンドを使用します。

MKDIR　PGM ⏎

のように入力します。このとき新しいディレクトリPGMがカレントディレクトリの下に作成されます。ここでDIRコマンドを実行すると、

PGM　<dir>　nn-nn-nn　nn:nn:nn

と表示されディレクトリが作成されたことが確認されます(<dir>の後ろには作成時の年月日と時刻が表示されます)。

ディレクトリとして付けられる名前は、ファイルに名前を付ける場合とまったく同じです。3．2節を参照してください。

すでに存在するディレクトリと同じ名前のディレクトリを作成することはできません。また、存在するファイル名と同じ名前のディレクトリも作成できません。このようなときは、

ディレクトリが作れません

と表示されます。

1つの階層構造に作成することのできるディレクトリの数や階層の深さの制限はありません。ディスクの容量が許す限り作成することができます。ただし、ルートディレクトリ上のファイル数（ディレクトリも含む）については制限があり、5インチ高密度（5″2 HD）フロッピーディスクの場合、192個まで作成できます。

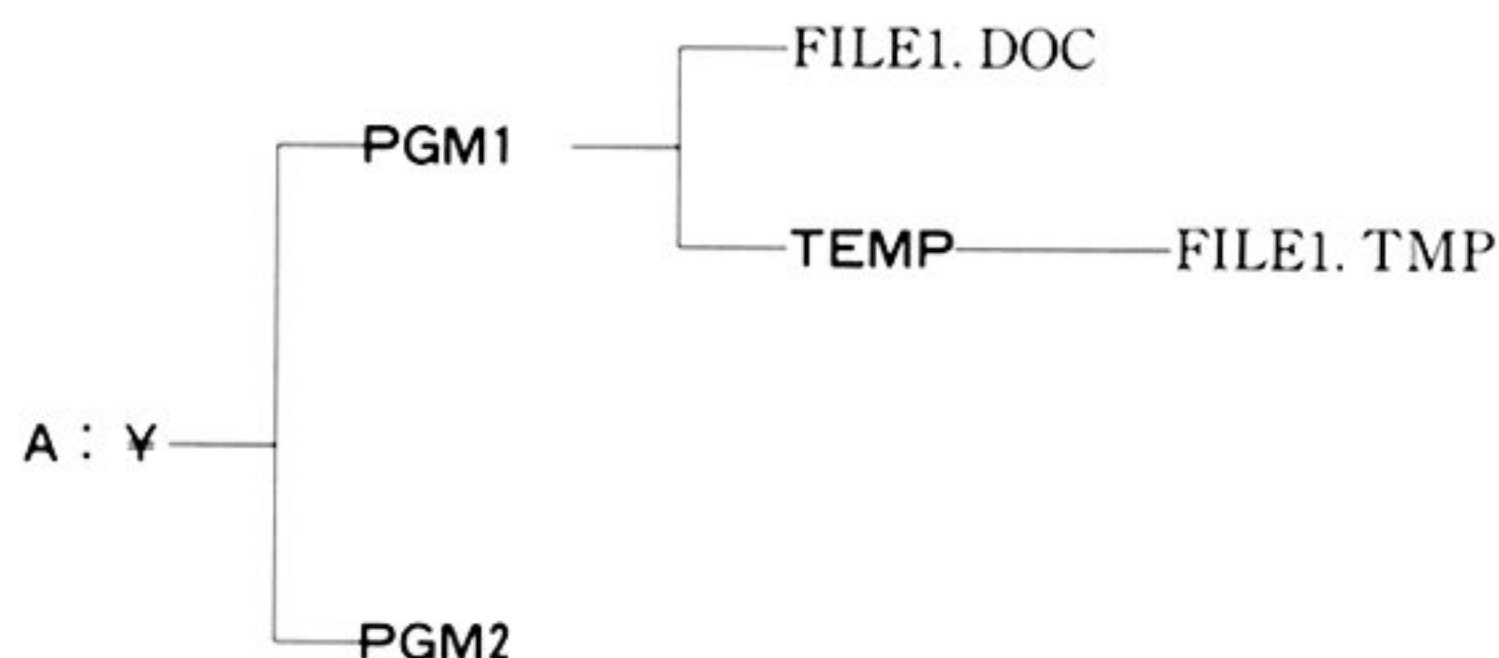
3．7．4　ディレクトリの削除

ディレクトリを削除するときは、RMDIR コマンドを使用します。

ディレクトリの作成は、MKDIR コマンドで簡単にできましたが、削除するときはそのディレクトリの下の階層にあるディレクトリやファイルをすべて削除しておく必要があります。

したがって、ディレクトリを削除するときは、階層の低いものから階層の高いものへと順番に削除していかなければなりません。

また、RMDIR コマンドは、自分自身のディレクトリと、そのすぐ上のディレクトリを削除することはできません。したがって、RMDIR コマンドは、自分自身のディレクトリのすぐ下のディレクトリを削除するときに使うのが一般的です。

たとえば、次のような階層構造でディレクトリ〝PGM1″を削除したいものとします（カレントディレクトリは、ルートであるとします）。図中、太字はディレクトリを表します。

まず、ディレクトリ〝TEMP″の下にある〝FILE 1．TMP″を削除しなければなりません。そのために、まずカレントディレクトリを〝TEMP″に移動してから DEL コマンドを使用します。

```
CHDIR  ¥PGM1¥TEMP ↵
DEL   FILE1.TMP ↵
```

次に、ディレクトリ〝TEMP″を削除します。自分自身のディレクトリを削除することはできませんから、1つ上のディレクトリに移動します。

```
CHDIR ..
```

ここでRMDIRコマンドを使って、

```
RMDIR TEMP
```

とします。まだこのままでは"PGM1"を削除できません。"FILE1.DOC"がこのディレクトリ中に残っているからです。

```
DEL FILE1.DOC
```

これで、"PGM1"中のファイルやディレクトリはすべて削除されました。これで初めて、目的のディレクトリを削除することができます。

```
CHDIR ..
RMDIR PGM1
```

なお、RMDIRコマンドを実行すると、ディレクトリに含まれるサブディレクトリやファイルが一度に消えてしまう、ということはありません。削除したいディレクトリの下の階層にサブディレクトリやファイルが存在しているときは、RMDIRコマンドは無効となります。

以上でファイルとディレクトリに関する解説を終了しました。ディレクトリの作成・変更・削除は、階層構造中におけるファイル操作を行う上で欠くことのできないものです。よく理解し、操作法を修得してください。なお、さらに詳しいことは、各コマンドの説明（第5章）をご覧ください。

第4章　コマンドについて

4.1　イントロダクション

コマンドは、コンピュータに処理を実行させるための言葉です。人間とコンピュータの間に位置していて、両者のコミュニケーションをはかるものです。

コマンドはふつうキーボードから入力されます。これを Human68k システムがコンピュータにわかる命令に翻訳し、それをコンピュータに伝達してさまざまな仕事を行わせるのです。

逆に言えば、コンピュータに何かの仕事をやらせようと思ったら、その仕事に応じたコマンドを選び、それをコンピュータに与えなくてはならないわけです。

Human68k のコマンドには、いろいろなものがあります。たとえば、次のような基本的な仕事をするためのコマンドがあります。

- **ディスクのフォーマット(初期化)、コピー、名前の付けかえ**
- **ファイルの表示、コピー、比較**
- **ディレクトリの表示、作成、変更、削除**
- **日付・時刻の設定**
- **プログラムの編集、実行**
- **周辺機器とのインターフェイスの管理**

これらのほかにも、実にさまざまな機能をもったコマンドがたくさんあります。これらのひとつひとつの働きを理解することにより、コンピュータを使いこなすことができる、というわけです。

それぞれのコマンドについては、第5章に、アルファベット順にコマンドを並べて詳しく説明していますので、そちらを参照してください。

4.2　コマンドの種類

本書では、Human68k の COMMAND. X の中に含まれているコマンドを内部コマンド、ディスク上に存在する実行可能なファイルを外部コマンドと分類しています。

●内部コマンド

コマンドプロセッサ (COMMAND. X) の内部にあります。カレントドライブがどこであっても、また、カレントディレクトリがどこであっても、つねに実行することができます。

●外部コマンド

コマンドプロセッサの外部 (システムディスク上の "BIN" というディレクトリの中)にあり、おのおの独立したファイルです。拡張子に ".X" または ".R" がついています。カレントドライブのカレントディレクトリ中にそのファイルがあればそのまま実行できますが、そうでなければ、そのファイ

ルにドライブ名とフルパス名を付けて実行しなければなりません。

ただし、PATH コマンドにより外部コマンドのファイルがどこにあるかをあらかじめ指定しておく、という便利な方法があります。これについては、「4．5　外部コマンドと PATH」の項に詳しく解説してありますので参照してください。

4．3　コマンドオプション

コマンドオプションとは、コマンドの実行時に与える Human68k への追加情報のことで、コマンドの後に続けて指定します。オプションを省略すると、ふつうはそのコマンドがあらかじめ用意している既定値（デフォルト値）が使われます。コマンドによってはデフォルト値がないものもあります。デフォルト値はコマンドによって異なりますから、くわしくは第５章のコマンド別の解説を参照してください。

オプションには、次のようなものがあります。

●d：

ドライブ名を指定します。ドライブ名を表す文字に続いてコロン（：）がつきます。

●ファイル名

コマンドに与えるファイル名です。拡張子付きのファイルを指定するときには、拡張子を省略することができません。

●拡張子

ファイル名拡張子です。コマンドによって省略可能です。

●パス名

階層ディレクトリ構造の中で、指定したファイルがどこにあるかを示します。次のような書式です。

〔¥〕〔〈ディレクトリ名〉¥〕…〔〈ディレクトリ名〉¥〕〈ディレクトリ名〉

●スイッチ

Human68k のコマンドを制限するもので、コマンドによって異なります。ふつう、スラッシュ（／）とアルファベット文字からなります。スラッシュ（／）の前には必ずスペースを入れてください。

例）　CHKDSK　B：　／A　（〝／A〟がスイッチ）

●パラメータ

コマンドの実行の際に必要な、細かい情報です。多くは、何種類かあるパラメータのうちの１つを選ぶ形で指定します。

例）　BREAK　ON　（〝ON〟がパラメータ）

ダブルクォーテーション（ ″ ）で囲まれた文字列は、その中にどんな文字が含まれていても、パラメータの一部として扱われます。

4．4　コマンドの共通事項

次に示すのは、Human68kの全コマンドに共通する事項です。

(1)　コマンドの多くは、1文字以上のオプションとともに使用します。

(2)　コマンドの入力は、そのコマンド名が大文字であっても小文字であっても、大文字、小文字、またはその混合で指定しても構いません。ただし、スイッチによっては大文字、小文字を区別する必要があるものもあります。

(3)　コマンドとオプションを区切る区切り記号（デリミタ）には、スペースやカンマを使うのがふつうです。そのほか、セミコロン、等号、タブ文字なども区切り文字として使用することがあります。

(4)　ファイル名の指定の中に区切り記号は入れられません。

(5)　メッセージとともに

何かキーを押してください

などと表示されたときは、メッセージに従って処置してください。

(6)　拡張子つきのファイル名を指定するときは、ふつう、拡張子を省略することはできません（コマンドの中には省略できるものもあります）。

(7)　コマンドは、リターンキーを押されるまでは処理を開始しません。

(8)　コマンドを実行中に［CTRL］＋［C］を押すと、ほとんどの場合はそのコマンドの実行を中止することができます。ただし、コマンドによっては中止できなかったり、または別の動作をすることがありますから、注意してください。

(9)　ワイルドカードやデバイス名は、コマンド名として指定することはできません。

(10)　ディスプレイへの出力が1画面以上に渡る場合、そのままにしておくと情報がどんどん画面からスクロールして流れ出てしまいます。このようなときは、［CTRL］＋［S］で画面のスクロールを止めることができます。任意のキーでスクロールを再開できます。

(11)　コマンド入力の際には、テンプレート機能を使用することができます。

(12)　コマンドモードを表すプロンプトは、PROMPTコマンドを改めて指定しない限り〝A>〟などの形をしています。このときの〝A〟はカレントドライブ名を表します。

4．5　外部コマンドとPATH

外部コマンドを実行するときは、実行したいコマンドファイルがカレントドライブのカレントディレクトリ中になくてはなりません。そうでない場合は、実行したいコマンドファイルにドライブ名とフルパス名を付けて指定する必要があります。

しかしこれでは、外部コマンドの所在位置をつねに覚えておかねばなりませんし、実行のたびにドライブやパス名を付けなければならないため入力もたいへんです。もし、階層の深いところにファイル名があったりしたら、ディレクトリ名をいくつも指定することになり、それこそたいへんな手間がかかってしまうことになります。

そこで、Human68kでは、〝PATH〟の設定により、これを解決しています。PATHの設定は次

のようにして行います。

PATH 〔<d：>〕<パス名> ⏎

ここで、<d：><パス名>には、実行したい外部コマンドファイルを収納したドライブ名およびディレクトリ名を指定します。

こうしてPATHを設定しておくと、入力されたコマンドはまず最初にカレントディレクトリ内でさがされ、もしそこに該当のコマンドファイルがないと、こんどは自動的にPATHで設定されたパス名のディレクトリ内でさがされます。

つまり、外部コマンドも、内部コマンドとまったく同じように、どこでも使うことができるようになるわけです。

PATHは、複数指定することもできます。

PATH　A：¥BIN；A：¥LOCAL；B：¥BIN ⏎

のように、セミコロン（；）で区切って並べてください。この場合、外部コマンドは、

カレントディレクトリ→A：¥BIN→A：¥LOCAL→B：¥BIN

の順にさがされるようになります。

PATHを設定した後で、設定したPATHを見たいときは、

PATH ⏎

とだけ入力します。現在設定されているPATHが、

path＝A：¥BIN；A：¥LOCAL；B：¥BIN

のように表示されます。

PATHは、いったん設定したら、またPATHコマンドで再設定を行うか、システムをリセットするか、電源を切るか、のどれかをしない限り有効です。

また、PATHは、外部コマンドファイルだけでなく、バッチファイルについても有効です。PATHに指定されたディレクトリ中にバッチファイルがあれば、これもさがされます。

システムの起動時は、PATHは設定されていません。したがって、自分のシステムディスク中の自動実行バッチファイル(AUTOEXEC.BAT)内に、このPATHコマンド行を入れておくと、起動後自動的にPATHが設定されるため、便利です。自動実行バッチファイルについては、次節を参照してください。

4.6　バッチ処理

よく使う一連のコマンドを、入力するのと同じイメージでテキストファイルの中に収めておくのが〝バッチファイル〞です。コマンドとしてそのバッチファイルを指定すれば、ファイルの中に書かれているコマンドを、そのつどキーボードから入力しなくても書かれた順に実行できます。このような処理の仕方は「バッチ処理」と呼ばれ、コンピュータの一般的な処理方法のひとつです。バッチファイルは、拡張子〝.BAT〞を付けて作成しなければなりません。ただし、実行の際はこの拡張子を省

略して入力できます。

バッチファイルは、付属のスクリーンエディタ ED か、日本語マルチフォントエディタなどで作成しますが、「3.5.2　ファイルの作成」のところで説明した、COPY コマンドを利用して作ることもできます。

ここでは、この COPY コマンドを利用して〝WORD. BAT〟という名前のバッチファイルを作ることにします。

COPY　CON　WORD. BAT ↵

と入力してください。すると画面上で改行して、入力を待っている状態になります。ここでバッチファイルの内容を入力してください。各行の末尾では、必ずリターンキーを押します。

VOL ↵
DIR　＊.TXT ↵
ED ↵
・
・
・

入力誤りは、いま入力している行内であれば BS キーで消して書き直せばよいのですが、いったんリターンキーを押してしまった行は訂正することができません。このときは、COPY コマンドを終了してファイルになった段階でスクリーンエディタや日本語マルチフォントエディタを使って修正してください。

バッチファイルの内容をすべて書き終わったら、CTRL ＋ Z を押します(CTRL キーを押しながら Z キーを押す)。続けてリターンキーを押します。COPY コマンドが終了して、〝WORD. BAT〟という名前のファイルができます。

これで、バッチファイル〝WORD. BAT〟ができあがりました。このバッチファイルを実行するには、

WORD ↵

と入力します。さきほど書いた、一連のコマンドがつぎつぎと自動的に実行されていきます。

なお、バッチファイルを使用する際は、次のような規則、注意点を守らなくてはなりません。

(1) バッチファイル中の1行の大きさは、255文字までです。超えた行があると、バッチ処理が中断されます。

(2) バッチファイル中にバッチファイルを実行するよう指定すると、そのバッチファイルに実行が移った時点で前のバッチファイルは無視されてしまい、それより後に書いておいたコマンドは実行されないままコマンドモードにもどってしまいます。このような場合は、／C スイッチ付きの COMMAND コマンドを実行するようにしてください(詳しくは第5章の COMMAND コマンドの項を参照してください)。

(3) バッチファイルの実行が終了すると、ECHO モードは強制的に〝ON〟になります。

4.6.1　パラメータを用いたバッチファイルの作り方

バッチファイルにパラメータを使うと、異なるデータに対しても汎用的に使えるバッチファイルを作ることができます。Human68kでは、バッチファイルの中に「仮パラメータ」を入れておき、実行時にコマンドオプションとして実際のパラメータ(「実パラメータ」)を指定することができます。

仮パラメータ名は、%記号を付けた1桁の番号（%0、%1……、%9）で指定します。

次に示すのは、仮パラメータを用いたバッチファイル〝NAME. BAT〞の内容例です。

```
COPY  %1   TEMP.TMP
COPY  %2  %1
COPY   TEMP. TMP   %2
DEL    TEMP. TMP
```

これを実行するには、実パラメータを付けて、たとえば次のように入力します。

```
NAME    FILE. 001   FILE. 002 ⏎
```

この場合、%1には〝FILE. 001〞が、%2には〝FILE. 002〞がそれぞれ代入され、次のようなコマンドをおのおの実行したのと同じになります。

```
COPY    FILE. 001   TEMP. TMP
COPY    FILE. 002   FILE. 001
COPY    TEMP. TMP    FILE. 002
DEL    TEMP. TMP
```

したがって、結果的に、〝FILE. 001〞の内容と〝FILE. 002〞が交換されます。

パラメータを用いたバッチファイルを使用する際の規則、注意点を、以下にあげます。

(1) 仮パラメータ〝%0〞は、つねに指定したバッチファイル自身のファイル名に相当します（拡張子も含まれます)。

(2) バッチファイルの実行時には、仮パラメータ〝%1〞は1番目のコマンドオプション（実パラメータ）に、仮パラメータ〝%2〞は2番目のコマンドオプション（実パラメータ）に対応します。以下、順に仮パラメータの番号に実パラメータの順番が対応します。

(3) 仮パラメータの数よりも実パラメータ（コマンドオプション）の数が多いときは、余分な実パラメータは無視されます。逆に、実パラメータの方が少ないときには、足らない分に対応する仮パラメータにはヌル文字が適用されます（何も代入されない)。

(4) 仮パラメータは、10個まで使用できます(%0～%9)。10個以上必要な場合は、バッチコマンドのSHIFTコマンドを使用します。

(5) バッチファイル内では、仮パラメータを表現するのに%記号を使いますので、そのまま%だけを単独で使うことはできません。もし%記号を文字として使いたければ、%%と2つ重ねて指定します。

(6) バッチファイルに与えるパラメータには、文字〝^@〞（コード$00）を入れてはいけません。それ以降の文字は無視されます。

4.6.2 自動実行バッチファイル(AUTOEXEC.BAT)の作り方

バッチファイルの特殊なかたちとして、自動実行バッチファイルというものがあります。これは、システムを起動したときに、自動的に実行されるようになっているものです。

これを利用すると、次に示すような、システム起動のたびに入力しなければならないような定型的な処理とか、環境の設定とかをキーボードからそのつど行うことなく、自動的に行うことができます。

- **起動後、すぐX-BASICを立ち上げる**
- **よく使うコマンドファイルやデータファイルを RAM ディスクにコピーする**
- **PATH を設定する**

などなど……

この自動実行バッチファイルを作るのは、とても簡単です。前項で述べた、通常のバッチファイルと、その作り方はまったく同じです。ただ、そのバッチファイルの名前を、

AUTOEXEC. BAT

と付けるだけでいいのです。これはいわばシステムに予約された特別なファイル名で、その意味は、〝AUTOmatic EXECution BATch〟、すなわち〝自動実行バッチ〟です。

Human68k が起動すると、起動に必要なファイル類をすべて読み込んだ後、システムはディスクファイル上からこの〝AUTOEXEC. BAT〟をさがします。そしてもしこのファイル名をもつファイルがあると、システムはすぐにその自動実行バッチファイルを読み、実行します。

次に、自動実行バッチファイルの典型的な例を示します。これは、Human68k システムディスク用の環境設定を行うものです。

```
ECHO    OFF
ECHO    Human68k  システムディスク
VERIFY   ON
BREAK    ON
PATH    A:¥;A:¥BIN;A:¥LOCAL;B:¥
```

4．7　コマンド行で利用できる機能

Human68kでは、ふつうはキーボードから入力を行い、画面に出力します。これはキーボードが「標準入力」になり、画面が「標準出力」になっているからです。

Human68kでは、コマンド行中に簡単な記号を指定することにより、これらの標準入力と標準出力のデバイスを変更して、画面ではなくプリンタに出力したり、ディスクのファイルから入力したりすることができます。これを「入出力のリダイレクト機能」と呼びます。

同時に、コマンド行中に簡単な記号を指定するだけで、あるコマンドの実行結果を、そのまま他のコマンドの標準入力として直結することのできる「パイプ機能」もあります。

さらに、1つのコマンド行中に、簡単な記号とともに複数のコマンドを一度に指定し、これを順に実行させることのできる「マルチ処理機能」があります。

4．7．1　リダイレクト機能

多くのコマンドは、画面を標準出力としていますから、なにも指定しなければ出力は画面に表示されます。

たとえば、次のコマンドはカレントドライブのカレントディレクトリのディレクトリ内容を画面に表示します。

DIR ⏎

しかし、次のように指定すれば、画面に出たのと同じイメージでプリンタにディレクトリ内容を印字できます（ただし、このとき画面にはディレクトリ内容は表示されません）。ここで〝>〟は、不等号の〝大なり〟記号を表します。

DIR　> PRN ⏎

このように、〝>〟を使うことにより、標準出力に出されるべきデータを、ファイルや他のデバイスにそのまま転送することができるのです。これが、出力のリダイレクトです。

ここで指定した標準出力は、デバイスファイルである〝PRN〟、すなわちプリンタですが、次のようにすれば、コマンドの出力内容をファイルとして作成することができます。

DIR　> FILES. DAT ⏎

Human68kは自動的にこの〝FILES. DAT〟を作り、ディレクトリ内容を書き込みます。このとき、すでにあるファイル名を指定しても警告は出ず、そのままファイルの内容を書き換えてしまいますから注意してください。

新しいデータを、すでにあるファイルの末尾に付け加える（アペンドする）こともできます。このときは、出力のリダイレクト記号である不等号〝>〟を、2つ続けて書きます。次の例では、現在ある〝XFILES.DAT〟の内容に、いまのディレクトリ内容が付け加えられます。

DIR　>> XFILES. DAT

リダイレクト機能は標準出力だけでなく、標準入力も変更することができます。このときリダイレ

クトを受けるコマンドは、標準入力であるキーボードからの入力を必要としているコマンドでなければなりません。入力のリダイレクトは、不等号の“<”（小なり）を使用します。
　たとえば、

　　SORT　<　FILES. DAT ↵

では、ディレクトリ内容の入ったデータファイルである“FILES. DAT”の内容をそのまま入力として、SORT コマンドが働きます。このような入力ファイルがすでに存在するなら、1行ずつキーボードから入力するよりはずっと簡単に済みます。
　ただしこのときは、SORT コマンドによって並べかえられた出力は、標準出力である画面に表示されます。出力が長いとすぐに画面からスクロールしていって消えてしまいます。そこで、この出力をファイルにとっておくようにするには、出力のリダイレクトを組み合わせて次のようにします。

　　SORT　<　FILES. DAT　>　FILES. SRT ↵

　並べかえられたディレクトリ内容は、FILES. SRT というファイルに収納されます。

4.7.2　パイプ機能

　標準入力からデータを読み込み、標準出力に出力するように作られたコマンドを、「フィルタ」と呼びます。Human68k では、

　　FIND
　　MORE
　　PR
　　SORT

の4つのコマンドがフィルタです。
　さて、ディレクトリ内容をアルファベット順に並べかえて表示した場合、出力のリダイレクト機能を使っていったんファイルに書き込み、それを入力のリダイレクト機能で SORT することができることは、前項で解説しました。しかしこの方法だと、間にひとつ余分なファイルが残ってしまいます。
　このようなときは、SORT コマンドがフィルタであることを利用して、パイプ機能を使うと、途中で一時的なファイルを作成しなくても、2つのコマンド間を直結してくれます。したがって2つのコマンド行が必要なところを、1行のコマンドを入力するだけで済みます。
　パイプ機能は、コマンドとコマンドとの間を縦線記号“|”で区切って表現します。たとえば、

　　DIR | SORT ↵

は、ディレクトリ表示（パイプ記号の左側）をそのまま SORT コマンド（パイプ記号の右側）の標準入力とし、画面に表示します。この方法によれば、前項の例のような“FILES. DAT”というファイルを作ることなく、ディレクトリ内容をソートして見ることができるわけです。
　ソート結果を画面単位で表示したければ、1画面ごとの表示をしてくれる MORE コマンドを使って、

```
DIR | SORT | MORE ⏎
```

とすればよいでしょう。また、次のようにすれば、

```
DIR | SORT > PRN ⏎
```

ソート結果をプリンタに印字できます。

もちろん、ドライブ名やパス名、ファイル名を指定すれば、指定のファイルに書き込むこともできます。

```
DIR | SORT > B:¥DATA¥DIRNAMES.DAT ⏎
```

このように、入力のディレクトリ、出力のリダイレクト、およびパイプは、自由に組み合わせて使うことができます。

4.7.3　マルチ処理機能

マルチ処理とは、1行のコマンド行入力で複数のコマンドを実行するためのものです。これを行うには、コマンドとコマンドとの間を、縦線記号（|）2つで区切って並べます。

たとえば、

```
DIR || DATE || TIME ⏎
```

と入力すると、DIRコマンド、DATEコマンド、TIMEコマンドが順に実行されます。

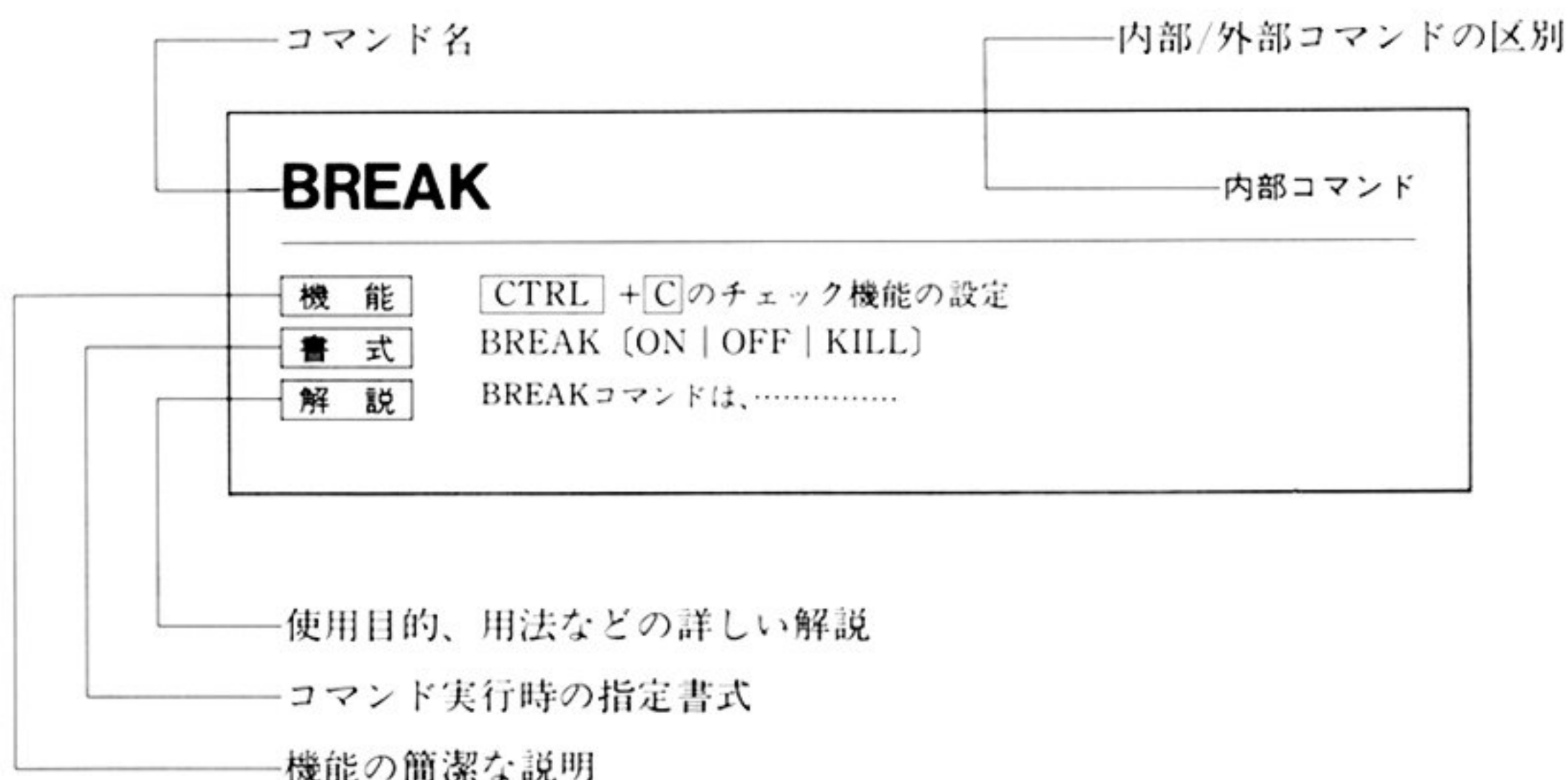
この機能を利用することにより、他のアプリケーションプログラムからCOMMAND.Xの内部コマンドを実行する場合、コマンドを実行するたびにアプリケーションプログラムに戻ることがなくなりますから、1回のコマンド指定で済ませることができます。

第5章　コマンド

5.1　イントロダクション

Human68kの各コマンドについて解説します。コマンドはアルファベット順に並んでいます。バッチ処理で使用するコマンドについては、5.3節でまとめて解説しています。

5.2　Human68kのコマンド

この節では、各コマンドを、次のような形式で解説しています。

内部／外部コマンドの区別について、詳しくは4.2節をご覧ください。また、書式の表記法については、(3)ページ「本書で用いる表記法」をご覧ください。

次に、各コマンドを機能別に分類しておきます。

●コマンドの機能による分類

システム関係

BREAK	CTRL + C のチェック機能の設定
CACHE	CPUのキャッシュのON/OFFを設定
CLS	ディスプレイの消去
COMMAND	コマンドプロセッサ COMMAND. X の起動
CTTY	入出力デバイスの変更
DATE	日付の表示・設定
EXIT	子プロセスとして起動された COMMAND. X からの親プロセスへのリターン

FASTIO	I/O関連の高速化
FASTOPEN	ファイル名の管理環境設定
FASTSEEK	デバイスの管理領域サイズの設定
HIS	ヒストリ行列の表示
KEY	ファンクションキーなどの設定
MEMFREE	使用可能なメモリサイズの表示
PROCESS	プロセスの情報を表示
PROMPT	プロンプトの設定
SCREEN	画面モードの設定
SET	環境文字列の値の設定
SPEED	RS−232Cインターフェイスに対するパラメータの設定・起動
SUBST	仮想ドライブの割り当ての設定・解除
SWITCH	メモリスイッチの設定
TIME	時刻の表示・設定
TIMER	時刻の表示・アラームのセット
VER	システムのバージョンの表示

ディスク関係

BACKUP	ハードディスクからフロッピーディスクに、1つまたはそれ以上のファイルのバックアップ作成
CHKDSK	ディスクの状態の報告
COPY2	ハードディスク上のファイルのフロッピーディスクへの退避・フロッピーディスク上の退避ファイルのハードディスクへの復帰
DISKCOPY	ディスク全体のコピー・比較照合
DRIVE	ドライブの種類、状態の表示・ドライブ名の交換
FORMAT	ディスクのフォーマット（初期化）
RECOVER	不良クラスタを含むファイルまたはディスクの修復
RESTORE	BACKUPコマンドでバックアップされたファイルの復元
SYS	Human68k システムの転送
VOL	ディスクのボリュームラベルの表示・変更

ディレクトリ関係

CHDIR(CD)	カレントディレクトリの変更・表示
DIR	ディレクトリ（ファイル名一覧）の表示
MKDIR(MD)	新しいディレクトリの作成
PATH	外部コマンド検索のためのPATHの設定
RMDIR(RD)	ディレクトリの削除
TEMP	テンポラリファイルのパスの設定
TREE	ディレクトリ構造のツリー形式での表示

WHERE	ファイルの検索

ファイル関係

ATTRIB	ファイルの属性の設定・解除
BIND	オーバーレイXファイルの作成・変更
COPY	ファイルのコピー・連結
COPYALL	ファイルおよびディレクトリのコピー
DEL(ERASE)	ファイルの削除
DUMP	16進表現・文字によるファイル内容の表示
FC	ファイル内容の比較
MOVE	ファイルの移動
REN (RENAME)	ファイル名の変更
TYPE	ファイル内容の表示
VERIFY	ベリファイ機能の設定

フィルタ関係

FIND	ファイル中の指定文字列の検索
MORE	1画面ごとの表示
PR	ファイル内容のページング
SORT	データのソート (並べかえ)

バッチ処理関係

ECHO	バッチ処理中におけるコマンド行表示の設定・メッセージの表示
FOR	コマンドの反復実行
GOTO	バッチ処理の流れの変更
IF	条件設定によるバッチ処理
PAUSE	バッチ処理の一時停止
REM	バッチ処理中におけるコメントの表示
SHIFT	バッチ処理中におけるパラメータのシフト

日本語処理関係

USKCGM	外字の作成・登録・削除

ATTRIB

外部コマンド

機能　ファイルの属性の設定・解除

書式　ATTRIB 〔+R｜−R〕〔+H｜−H〕〔<ファイル名>〕

解説　ATTRIBコマンドは、ファイルを誤って削除したり、別のファイルに上書きしてしまうといったミスを防ぐために使用します。また、ファイルに対し、不可視属性(DIRコマンドなどで見たり操作したりすることができないという属性) を与えるためにも使います。

●ファイルの属性を、読み出し専用に設定します

```
ATTRIB  +R  MSTFL.100 ↵
```

ファイルMSTFL. 100の属性が、読み出し専用に設定されます。DIRコマンドでファイル名を表示することはできますが、削除したり、内容を変更したりすることはできなくなります。

●ファイルの読み出し専用属性を解除します

```
ATTRIB  -R  MSTFL.100 ↵
```

ファイルMSTFL.100の読み出し専用の属性が、解除されます。実行後、ファイルは書き込み可能となります。

●ファイルの属性を、不可視属性に設定します

```
ATTRIB  +H  MSTFL.100 ↵
```

ファイルMSTFL. 100の属性が、不可視属性に設定されます。ファイルはDIR、DEL、TYPEなどのコマンドによる通常の操作では、取り扱うことができなくなります。

●ファイルの不可視属性を解除します

```
ATTRIB  -H  MSTFL.100 ↵
```

ファイルMSTFL. 100は、DIRコマンドで見ることができるようになり、その他のコマンドによる通常の操作も可能となります。

●ファイルの属性を表示します

ATTRIB　WKFL. 100 ⏎

ファイル WKFL. 100の現在の属性を表示します。現在のファイルの属性の設定状態を見るときに使用します。

ATTRIB　*. * ⏎

DIR コマンドで見えないものも含め、すべてのファイルの属性を表示します。

この書式では、ファイルの属性が各ファイル名の左側に、次のような記号で表示されます。

記号　属性名
A …… アーカイブ（通常のファイル）
D …… ディレクトリ
V …… ボリュームラベル
S …… システム
H …… 不可視
R …… 読み出し専用

なお、ATTRIB コマンドで設定・解除ができるのは、H および R のみです。

●ヘルプ画面の表示

ATTRIB ⏎

ATTRIB コマンドの使用方法を表示します。

BACKUP

外部コマンド

機　能

ハードディスクからフロッピーディスクに、1つまたはそれ以上のファイルのバックアップ作成

書　式

BACKUP　〈hd：〉〔〈パス名〉〕〔〈ファイル名〉〕〈fd：〉〔／S〕〔／A〕〔／D：〈日付〉〕〔／T：〈時刻〉〕

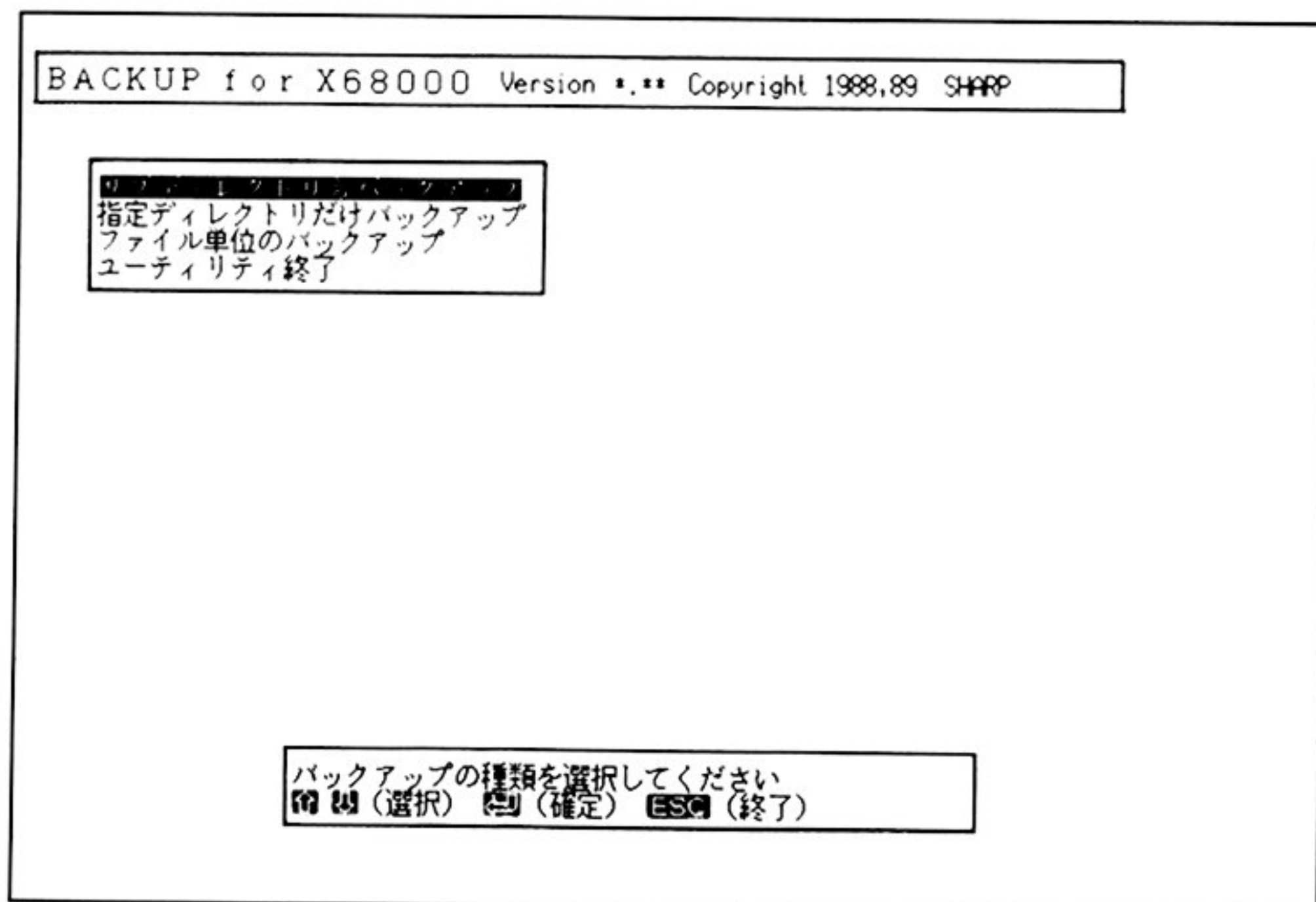
解　説

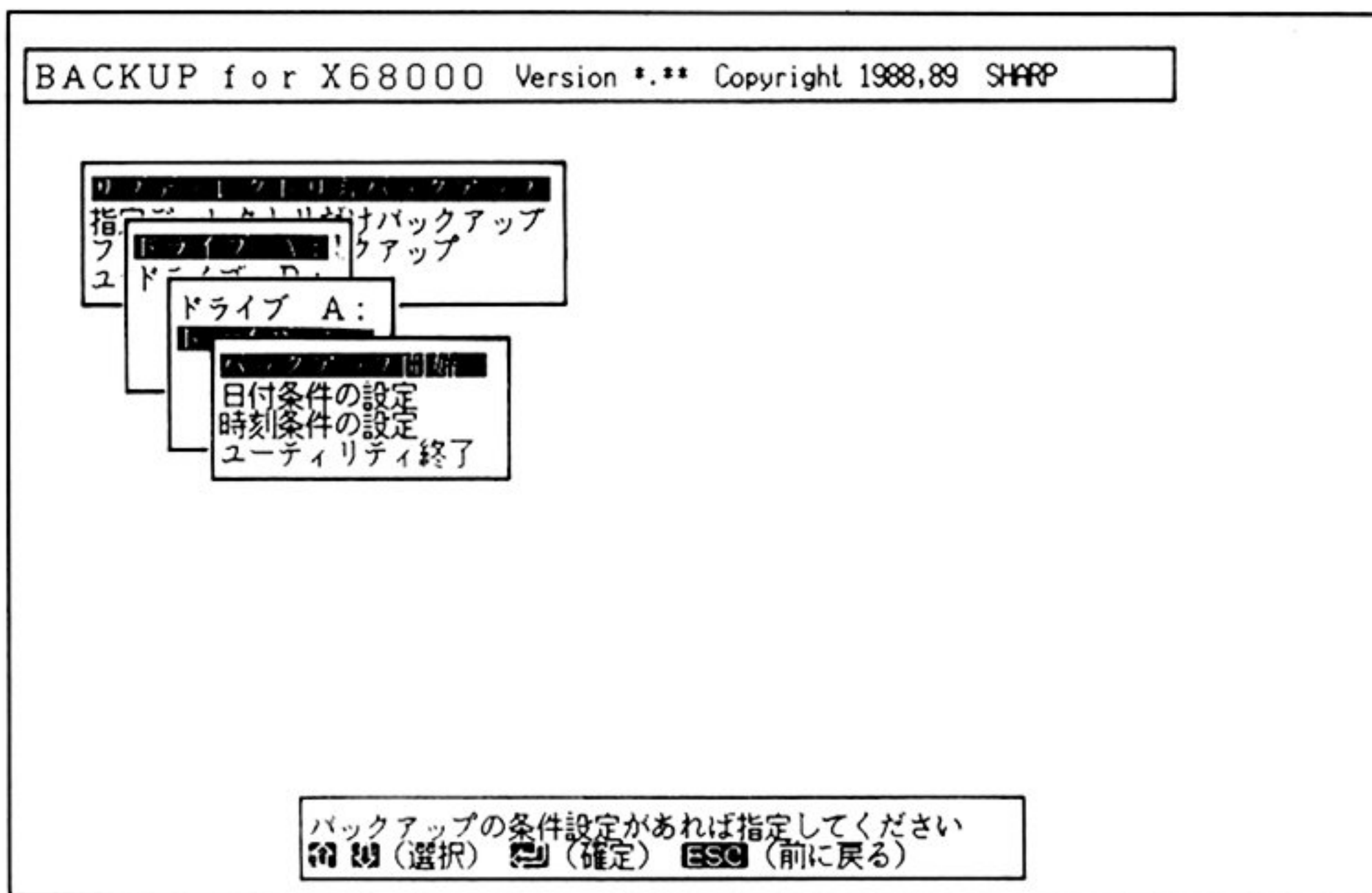
ハードディスクは大量のデータを格納できるという利点がある反面、一度事故が起きると失うデータも大量であるという欠点もあります。このため、定期的にバックアップをとり、データの保全に努める必要があります。

1．メニュー画面からの実行

BACKUP [↵]

と入力すると、次のようなメニュー画面が表示されます。

```
BACKUP for X68000  Version *.** Copyright 1988,89 SHARP

  指定ディレクトリだけバックアップ
  ファイル単位のバックアップ
  ユーティリティ終了

バックアップの種類を選択してください
[↑][↓]（選択） [↵]（確定） [ESC]（終了）
```

ここで、カーソルキーの [↑] [↓] で処理を選択して [↵] を押します。この後は画面に表示されるメッセージにしたがってドライブ名やディレクトリ名などを選択してください。

たとえば、ドライブA（ハードディスク）の内容すべてを、ドライブB（フロッピーディスク）にバックアップする場合は、次の画面のように〝サブディレクトリもバックアップ〟→〝ドライブA：〟→〝ドライブB：〟→〝A：¥〟→〝バックアップ開始〟を[↵]で選択します。

必要であれば最後に日付条件（指定した日付の間に作成したファイルをバックアップ）や時刻条件（指定した時刻の間に作成したファイルをバックアップ）を設定し、「バックアップ開始」を選択してください。

2．コマンド行からの実行

コマンド行で 書式 に従ったパラメータを設定した場合、メニュー画面に入らずにコマンドを実行します。

最初の〈hd：〉〈パス名〉〈ファイル名〉には、バックアップするハードディスクドライブのディレクトリまたはファイルを指定します。〈パス名〉〈ファイル名〉が省略された場合は、ルートディレクトリに存在する全ファイルが指定されたことになります。次の〈fd：〉には、バックアップされたファイルが作成されるフロッピーディスクドライブを指定します。

BACKUP コマンドでは以下のスイッチが使用できます。

／S　サブディレクトリもバックアップします。

／A　すでにバックアップされたフロッピーディスク上にある、バックアップされたファイルに追加します。古いファイルを削除しません。

／D　指定された日付以降に変更されたファイルをバックアップします。

／T　指定された時刻以降に変更されたファイルをバックアップします。

BIND

外部コマンド

機　能　オーバーレイXファイルの作成・変更

書　式　BIND　〔／O〕<作成するファイル名> <ファイル名>〔<ファイル名> …〕〔／D｜／X｜／L〕〔／T<パス名>〕

解　説

拡張子が“.X”になっている複数のXファイルを結合して、ひとつのオーバーレイXファイルを作成します。オーバーレイというのは、使用可能なメモリサイズに対して、プログラムやデータのサイズの方が大きいときに利用されるものです。

たとえば、TEST.X というプログラムを実行するためには800Kバイトのメモリが必要というときに、使用可能なメモリが500Kバイトしかない場合、メモリ不足でこのままでは実行できません。そこで、プログラムを作成するときに、ひとつのメインプログラムと複数のサブプログラムに分けて作成し、実行するときは、必要なサブプログラムをそのつどディスクから読み込んで実行するようにします。こうするとメモリが少なくても大きなプログラムを実行することができるようになります。

ところが、このようなプログラムを作るためには、メインプログラムでメモリの管理やディスクからの読み込み、実行などを行わなければならず、プログラミングがたいへんめんどうになります。

BINDコマンドは、このようなオーバーレイ構造のプログラムの作成を援助するコマンドです。複数のプログラムをオーバーレイにしたいときは、通常のプログラムを作成するときのように単にXファイルにすればよいだけです。BINDコマンドは以下のように使用します。

●複数のXファイルを結合して新しいオーバーレイXファイルを作成する

```
BIND  /O  TEST  TEST1  TEST2  TEST3 ↵
```

この例では、TEST1.X、TEST2.X、TEST3.X というXファイルから TEST.X というオーバーレイXファイルを作成します。拡張子は省略しても自動的に“.X”であると判断されます。

●すでにオーバーレイXファイルが存在するときに、別のXファイルを追加する

```
BIND  TEST  TEST4  TEST5 ↵
```

この例では、TEST.X というオーバーレイXファイルにさらに TEST4.XとTEST5.Xの２つのXファイルを追加します。

●オーバーレイXファイルの中にどのようなXファイルがあるか調べるとき

```
BIND TEST /L ↵
```

この例では、TEST.XというオーバーレイXファイルに含まれるXファイルを表示します。

●オーバーレイXファイルから特定のXファイルを削除する

```
BIND TEST TEST2 TEST4 /D ↵
```

この例では、TEST.XというオーバーレイXファイルからTEST2.XとTEST4.Xの2つのXファイルを削除します。

●オーバーレイXファイルから特定のXファイルを取り出す

```
BIND TEST TEST1 TEST3 /X ↵
```

この例では、TEST.XというオーバーレイXファイルからTEST1.XとTEST3.Xの2つのXファイルを取り出します。

●テンポラリファイルのパス名をドライブCに指定して、オーバーレイXファイルを作成する

```
BIND TEST TEST2 TEST4 /T C:¥↵
```

この例では、BINDコマンドを実行するときに作成されるテンポラリーファイルをドライブCのRAMディスクに指定して、TEST.XというオーバーレイXファイルを作成します。フロッピーディスクに空き容量がないときや、実行速度を速めたいときには、テンポラリーファイルをRAMディスクに設定すると便利です。

オーバーレイXファイルのロードと実行の方法は次のとおりです（別売の「C Compiler PRO-68K」が必要です）。

DOSコールの_execファンクションコールで、MDの下位バイトに0（ロードおよび実行）、1（ロードのみ）、3（特殊なロード）、MDの上位バイトにモジュール番号を与えてコールします。通常のXファイルの場合、モジュール番号は0であり、オーバーレイXファイルでも最初に結合したファイルが0、次のファイルが1……となります。詳しくは、「C Compiler PRO-68K」のマニュアルを参照してください。

● TEST.Xの中にTEST1.X、TEST2.X、TEST3.Xが入っているときにそれぞれをロードおよび実行するときは、

TEST 1 .XならばMDは$0000
TEST 2 .XならばMDは$0100
TEST 3 .XならばMDは$0200

となります。したがって、

TEST ⏎

と入力したときは、TEST1.Xがロードおよび実行されることになります。ですから、TEST1.Xでは_execファンクションコールを利用して、TEST2.XやTEST3.Xをロードおよび実行するように作成すればよいわけです。

● TEST.Xの中にTEST1.X、TEST2.X、TEST3.Xが入っているときにそれぞれをロードするときは、

TEST 1 .XならばMDは$0001
TEST 2 .XならばMDは$0101
TEST 3 .XならばMDは$0201

となります。

また、次のようなエラーメッセージが表示されたときは、ディスクのあき領域が足りないためテンポラリーファイルが作れません。充分なあき領域のあるドライブなどにテンポラリーファイルを作るようにパス指定してください。パス指定は前述の／T〈パス名〉またはTEMPコマンドで行えます。

テンポラリーファイルが作れません

BREAK　　内部コマンド

機　能　CTRL ＋ C のチェック機能の設定

書　式　BREAK 〔ON | OFF | KILL〕

解　説

　BREAK コマンドは、アプリケーションプログラムの実行中、CTRL ＋ C のチェックを行うか否かを設定するために使用するコマンドです。

　BREAK を ON に設定すると、アプリケーションプログラムがディスクの入出力などの Human68k システム内部の処理を行うたびに、CTRL ＋ C キーが押されたか否かのチェックが、行われるようになります。この設定を行うとコンパイルやリンク中にもプログラムを中断することが可能になります。

　BREAK を OFF に設定すると、キーボード入力、ディスプレイ出力およびプリンタ出力のいずれかの内部処理が行われるときにのみ CTRL ＋ C のチェックが行われ、その他の処理ではチェックされなくなります。したがってこの状態では、内部処理が行われている限り、プログラムの中断を行うことはできません。通常はこの設定でアプリケーションプログラムを実行します。

　BREAK を KILL に設定すると、どんな場合でも中断できなくなります。どうしても中断したい場合は、本体のINTERRUPT スイッチを押してください。

　BREAK は、システムの起動時には、OFF に設定されています。

　現在の設定状況を表示したいときは、

```
BREAK ↵
```

とします。BREAK が OFF のときは、

```
break は<off>です
```

と表示され、ON のときは、

```
break は<on>です
```

と表示され、KILL のときは、

```
break は <kill> です
```

と表示されます。

CACHE

外部コマンド

機　能

CPUのキャッシュコントロールの設定

書　式

CACHE　{[on] [off] [def]}　[i] [d] [s]

解　説

CPU：MC68030に内蔵のキャッシュメモリを使用するかどうかを設定します。

on　キャッシュオン。内蔵キャッシュメモリが使用できます。

off　キャッシュオフ。内蔵キャッシュメモリを使用しません。

def　S-RAMに設定されている状態にもどします。
　i、d、sの指定は無視されます。

i　MC68030に内蔵されている命令キャッシュを指定します。

d　MC68030に内蔵されているデータキャッシュを指定します。

s　指定したキャッシュの状態をS-RAMに設定します。

CHDIR(CD)

内部コマンド

機　能　　カレントディレクトリの変更・表示

書　式　　CHDIR　〔<d：>〕〔<パス名>〕

別表記　　CD

解　説

カレントディレクトリを変更するときに使用します。

CHDIR コマンドは、指定ドライブのカレントディレクトリを変更するときに使用します。

カレントディレクトリは、ドライブごとに設定できるため、<d：>を指定することにより、カレントドライブを移動することなく他のドライブのカレントディレクトリを変更することができます。

<d：>を省略したときは、カレントドライブとみなされます。

ディレクトリの指定の方法には、パス名をルートディレクトリから指定する絶対指定と、カレントディレクトリから指定する相対指定の、２種類があります。

●絶対指定によるディレクトリの変更

カレントディレクトリが¥１部¥１課¥１係のとき、ディレクトリ¥１部¥１課¥２係に変更したいときは、

CHDIR　¥１部¥１課¥２係 ↵

とします。

●相対指定によるディレクトリの変更

カレントディレクトリが¥１部¥１課¥１係のとき、ディレクトリ¥１部¥１課¥２係に変更したいときは、

CHDIR　..¥２係 ↵

とします。ここで〝..″は、カレントディレクトリより１つ上のディレクトリ（ここでは１課）を意味します。

●カレントディレクトリの表示

カレントディレクトリを表示するときには、なにもパラメータを与えずに、

CHDIR ↵

とします。カレントディレクトリが¥１部¥１課¥２係であれば、

¥１部¥１課¥２係

と表示されます。

CHKDSK

外部コマンド

機　能　ディスクの状態の報告

書　式　CHKDSK　〔<d：>〕〔／A〕〔／V〕

解　説

CHKDSK コマンドは、<d：> で指定したドライブに挿入されているディスクのFAT 領域とディレクトリ領域を調べ、ファイルやディレクトリの使用状態などをチェックします。

コマンドを実行すると、次のようなステータスレポートを表示します。バイト数の表示はキロ（K）バイト単位です。

```
d：vvvvv は yyyy-mm-dd　hh：mm に作成されました
  nnnn K バイト：全ディスク容量
    nn K バイト： n 個のシステムファイル
    nn K バイト： n 個のディレクトリ
  nnnn K バイト： n 個のユーザーファイル
 nnnnn K バイト：使用可能ディスク容量
```

ここで、d：vvvvv は、ドライブ名とボリュームラベル名、yyyy-mm-dd は日付、hh：mm は時刻をそれぞれ表し、n は数値を表します。

スイッチを指定しないときは、FAT 領域をサーチして、ファイルのデータブロックが連続しているかどうかのチェックを行います。不連続なデータブロックをもつファイルがあると、次のようなメッセージをステータスレポートの前に表示します。

```
d：¥xxxxx　n 個の不連続ブロックがあります
```

ここで、d：¥xxxxx は、ドライブ名およびパス名付きのファイル名を表します。

もしスイッチが指定されていれば、ステータスレポートの前に、そのスイッチ指定した情報を表示します。スイッチには、次の2つを指定できます。

●／A

スイッチ／A を指定すると、ステータスレポートまたはエラーメッセージの他に、すべてのファイル（サブディレクトリを含む）を表示します。同時にそのファイルが収められているセクタの範囲が報告されます。

```
------　xxxxx. xxx　　$nnnn～$nnnn
```

ここで、------は属性を、xxxxx. xxx はファイル名を、$nnnn～$nnnn はセクタの報告を、それぞれ表します。

属性については、ATTRIB コマンドで表示されるものと同一です。ATTRIB コマンドを参照してください。

●／V

スイッチ／Vを指定すると、ステータスレポートまたはエラーメッセージの他に、すべてのファイル（サブディレクトリを含む）を表示します。ただし、セクタの状態は報告されません。

CLS

内部コマンド

機能　ディスプレイの消去

書式　CLS

解説　CLS コマンドは、画面上に表示された文字を消去（クリア）するときに使います。

CLS コマンドを実行すると画面の表示はすべてクリアされ、カーソルはホームポジションに戻ります。

何か作業を行う際に、表示された画面をクリアしたいときには、

CLS ↵

と実行すると、表示はすべてクリアされプロンプトが画面の左上にきます。

注意：CLS コマンドでクリアできるのはテキスト（文字）画面だけです。グラフィック画面には、何の影響も与えません。

COMMAND

外部コマンド

機　能

コマンドプロセッサ COMMAND. X の起動

書　式

COMMAND　〔／P〕〔／D〕〔／E：〈環境エリアサイズ〉〕〔／H：〈ヒストリエリアサイズ〉〕〔〔／C〕〈文字列〉〕

解　説

COMMAND コマンドは、ふつう、Human68k のコマンドプロセッサ COMMAND. X を子プロセスとして起動する際に使用します。

アプリケーションプログラムによっては、COMMAND. X を子プロセスとして起動することができるように作られたものがありますが、そのアプリケーションプログラム中から直接 COMMAND. X を呼び出すのではなく、コマンド行を指定して呼ぶことができるようになっているような場合、とくにこの COMMAND コマンドが意味を持ちます。

また、COMMAND コマンドは、CONFIG. SYS ファイルの中の〝SHELL=〟行中に指定する際に、最も一般的に用いられるものです。これについての詳細は、「第7章　システムの構築」を参照してください。

COMMAND コマンドは、パラメータを与えずに実行すると、現在の Human68k のバージョンを表示した後、子プロセスとして COMMAND. X をロードします。

COMMAND コマンドには、以下のようなパラメータ（スイッチなど）を付加することができます。これら以外のスイッチが指定されると、それ以降のコマンド行は無視されます。

●／P

COMMAND. X からそれ以上高いレベルに抜け出すことを防止するときに使用します。これを付けて COMMAND コマンドを実行すると、子プロセスとして起動された後、EXIT コマンドが実行されてもそのプロセスのレベルより上に抜け出せなくなります。

また、このスイッチが指定されると、COMMAND.X が読み込まれた後、自動的に AUTOEXEC.BAT ファイルが実行されます。

●／D

／P と同時に指定すると、AUTOEXEC. BAT ファイルの起動をしないようになります。

このスイッチは、CONFIG. SYSの〝SHELL=〟または／Pとともに指定します。単独で指定したときは、コマンドプロセッサCOMMAND. Xを子プロセスとして起動させます。

●／E：〈環境エリアサイズ〉

SET コマンド用に、〈環境エリアサイズ〉で指定された数の環境エリアを確保します。環境エリアは、1個当たり 256 バイトで、0～255 の範囲で指定できます。0 は

512バイトを意味します。デフォルト値は0（512バイト）です。

●／H：〈ヒストリエリアサイズ〉

HISコマンド用に、〈ヒストリエリアサイズ〉で指定された数のヒストリエリアを確保します。ヒストリエリアは、1個当たり256バイトで、0～255の範囲で指定できます。0は512バイトを意味します。デフォルト値は0（512バイト）です。

ヒストリの詳細については、HISコマンドの項を参照してください。

●／C〈文字列〉

文字列で指定されたコマンドがコマンドプロセッサにより実行され、終了後戻ります。これを利用すると、バッチファイル中から別のバッチファイルを呼び出すことができます。たとえば、

```
COMMAND  /C  TEST
```

では、バッチファイルの実行中にコマンドプロセッサが起動され、次にTEST.BATが呼び出されて実行された後、最初のバッチファイルの次の行の実行に戻ります。

ふつう、バッチファイルから別のバッチファイルを呼び出すことはできませんが、このスイッチを使用することにより、別のバッチファイルの実行が可能になります。

もちろん、外部コマンドや内部コマンドもこの形式で実行することができます。この場合、現在のコマンドプロセッサではなく、子プロセスとして実行されるCOMMAND.Xの下でコマンドが実行されることになります。

なお、／Cは省略し、〈文字列〉のみを指定しても同じ動作になります。

／Cの後ろに、別のオプションやスイッチをつけることはできませんので注意してください。

COMMAND.Xは、バッチファイルの実行中、および／C付きのCOMMANDコマンドを実行したときに、次のようなエラーコードを返します。このエラーコードは、IFコマンドのERRORLEVELオプションで利用することができます。

$00nn（0～255）

外部コマンドのプロセス終了コードと同一の値です。

$01nn（256～511）

下位バイトは外部コマンドのプロセス終了コードと同一の値ですが、そのコマンドが常駐終了した場合、この値となります。

$02nn（512～767）

コマンドが［CTRL］+［C］によって中断された場合、この値となります。下位バイトは不定となります。

＄03nn (768～1023)

ファンクションコールエラーにより強制終了（アボート）された場合、この値となります。下位バイトは Human68k のエラーコードとなります。

＄04nn (1024～1279)

バッチ処理の実行時エラーの場合、この値となります。下位バイトは不定となります。

＄05nn (1280～1535)

内部コマンド中でエラーが生じた場合、この値となります。下位バイトは次のとおりです。

00	パラメータエラー
01	ファイル読み出しエラー
02	ファイル書き込みエラー
03	ファイルが見つからないとき
04	ディレクトリがいっぱいのとき
05	ディスク容量がいっぱいのとき

＄06nn (1536～1791)

指定のコマンドが見つからない場合、この値となります。下位バイトは不定となります。

COPY

内部コマンド

機　能　ファイルのコピー・連結

書　式　COPY　{〈d：〉|〈パス名1〉|〈ファイル名1〉}　〔{〈d：〉|〈パス名2〉|〈ファイル名2〉}〕〔／V〕〔／Q〕〔／B〕

解　説　COPY コマンドは、ファイルやディレクトリをコピーするときに使用します。ファイルのコピーはファイルおよびディレクトリ単位で行うことができます。

スイッチ／Vをつけるとディスクをコピーしながら自動ベリファイを行います。すなわち、コピー後自動的に、コピー先のファイルやディレクトリのデータを読み込み、書き込みエラーがないかどうかの照合を行います。したがって、コピーの内容は保証されますが、実行の速度は遅くなります。複数のファイルをコピーする場合、スイッチ／Qをつけると、コピーの対象となるファイル名とともに確認のメッセージを表示してからコピーを行います。

バイナリファイルを連結する場合は、スイッチ／Bをつけてコピーを行います。

1．ファイル単位のコピー

●あるファイルを違うファイル名でコピーします

COPY　〈ファイル名1〉　〈ファイル名2〉 ↵

〈ファイル名1〉とまったく同じ内容を持つファイルが、同一ドライブ上に〈ファイル名2〉という名前で作成されます。

●ドライブ A にあるファイルをドライブ B に違う名前でコピーします

COPY　A：〈ファイル名1〉　B：〈ファイル名2〉 ↵

〈ファイル名1〉とまったく同じ内容を持つファイルが、ドライブ B 上に〈ファイル名2〉という名前で作成されます。

●ドライブ A にあるファイルをドライブ B に同じファイル名でコピーします

COPY　A：〈ファイル名〉　B： ↵

このとき、B：の後にファイル名を指定する必要はありません。

●ドライブ A のあるディレクトリ中のすべてのファイルを、ドライブ B にそれぞれ同じファイル名でコピーします

COPY　A：〈パス名〉　B： ↵

ドライブＡの<パス名>で指定されたディレクトリ中に存在するすべてのファイルが、まったく同じ名前でドライブＢのカレントディレクトリ上に作成されます。
このときＢ：の指定の後にファイル名を指定する必要はありません。

また、スイッチ／Ｑをつけると、コピーの対象となるファイル名とともに確認のメッセージが表示されたのち、コピーを行います。

COPY　A：<パス名>　B：　／Q ⏎

XXXX.XXX をコピーしますか <Y／N>
⋮

２．ディレクトリ単位のコピー

●ひとつのディレクトリにあるすべてのファイルを他のディレクトリにコピーします

COPY　<パス名１>　<パス名２> ⏎

<パス名１>で指定されたディレクトリ中に存在するすべてのファイルが、まったく同じ名前でそれぞれ<パス名２>で指定されたディレクトリ上に作成されます。なお、<パス名２>には、すでに存在しているディレクトリ名を指定しなくてはなりません。
このディレクトリ単位のコピーは、同一ドライブ上でも、異なるドライブ間でも行うことができます。

３．エラー

コピー元とコピー先のファイル名やディレクトリ名が同じときは、コマンドの実行が中止されます。自分自身にコピーすることになるためです。このとき、次のメッセージを表示します。

コピー元とコピー先が同一です

注意：ドライブ名、パス名を指定する際に、指定すべきドライブがカレントドライブである場合、および指定するパス名がカレントディレクトリである場合には、それぞれ省略することができます。

４．ファイルの連結

複数のファイルをひとつのファイルにまとめるときは、＋記号でファイル名をつないだり、ワイルドカードで複数のファイルを指定します。

●TEST1.DOC と TEST2.DOC を TEST.DOC にまとめます

```
COPY  TEST1.DOC+TEST2.DOC  TEST.DOC ⏎
```

●拡張子が“.DOC”のファイルをすべてALL.DOCとしてまとめます

```
COPY  *.DOC  ALL.DOC ⏎
```

上記のファイルの連結の例はテキストファイルのみ可能で、バイナリファイルの場合は、スイッチ／Bをつけなければなりません。

●TEST1.XとTEST2.XをTEST.Xにまとめます

```
COPY  TEST1.X+TEST2.X  TEST.X  /B ⏎
```

注意

ファイルをコピーするとき、コピー元やコピー先が、仮想ドライブや仮想ディレクトリを含むときは、次のことに注意してください。

コピー元の指定とコピー先の指定が異なっても、実体としては同じドライブやディレクトリになることがあります。たとえば、仮想ドライブの設定が、

```
C:=A:¥TEST
```

となっているときに、コマンド行で

```
COPY  A:¥TEST¥SAMPLE1.DAT  C: ⏎
```

と入力すると、COPYコマンドが実際に行う作業が

```
COPY  A:¥TEST¥SAMPLE1.DAT  A:¥TEST ⏎
```

となり、コピー元とコピー先がA：¥TESTと同じディレクトリになります。

このような場合は、コピーは正しく行われません。また、ファイルが消失する可能性もありますので、注意してください（仮想ドライブ、仮想ディレクトリについては、SUBSTコマンドをご覧ください）。

5．各デバイスへのファイル出力

●OPMファイルをFM音源やMIDIのデバイスから演奏するときや、PCMファイルをPCMデバイスから再生するときは、コピー先を出力するデバイス名にします

COPY　X68K.OPM　OPM ↵

また、OPMファイルをMIDIデバイスから演奏するときや、PCMファイルをPCMデバイスから演奏するときも、同様の書式で入力します。

COPY　X68K.OPM　MIDI ↵
COPY　SAMPLE.PCM　PCM ↵

なお、FM音源、MIDI、PCMなどのデバイスから演奏するときは、CONFIG.SYSファイルにOPMDRV3.Xが登録されている必要があります。詳しくは本書「7章　システムの構築」を参照してください。

DEVICE=¥SYS¥OPMDRV3.X　♯180　/P64　/OPM

●テキストファイルをプリンタで印刷するときは、コピー先をPRNまたはLPTにします

COPY　TEST.TXT　PRN ↵

なお、テキストファイルをプリンタで印刷するときは、CONFIG.SYSファイルにPRNDRV.SYSなどが登録されている必要があります。詳しくは本書「7章　システムの構築」を参照してください。

COPY2

外部コマンド

機　能

ハードディスク上のファイルのフロッピーディスクへの退避・フロッピーディスク上の退避ファイルのハードディスクへの復帰

書　式

(1)　退避　COPY2　〔<hd：>〕〔<パス名>〕<ファイル名>〔<fd：>〕

(2)　復帰　COPY2　〔<fd：>〕<ファイル名>〔<hd：>〕〔<パス名>〕／R

解　説

COPY2コマンドは、フロッピーディスク1枚に収まらないような大きなファイルも、複数枚に分けて退避することができます。フロッピーディスクには、データを詰められるだけ詰めますから、効率よく退避を行うことができます。

また、COPY2コマンドは、フロッピーディスクに退避したファイルをハードディスク上に復帰することもできます。

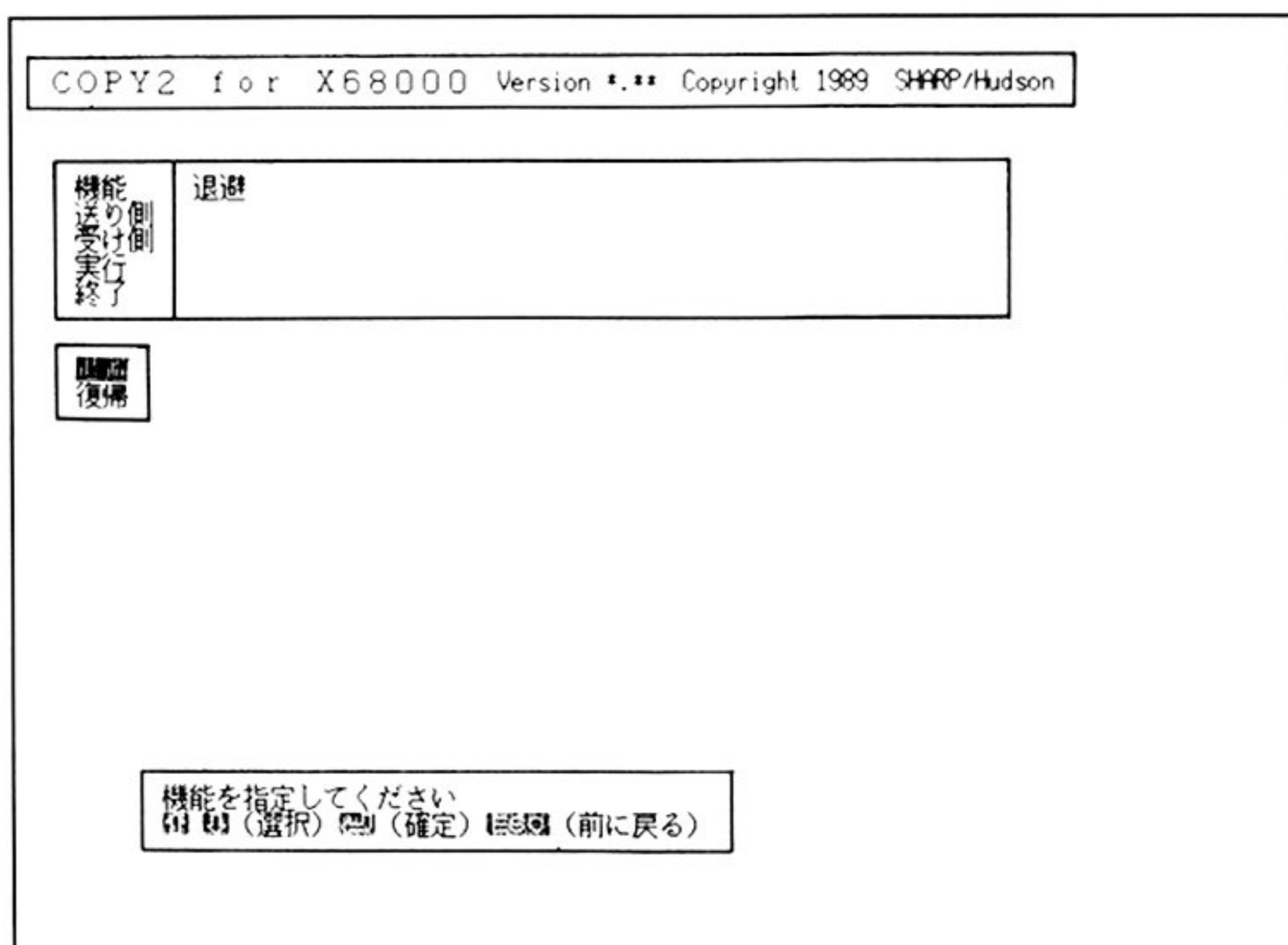
1．メニュー画面からの実行

COPY 2 ↵

と入力すると、画面は次のようなメニュー画面になります。

```
COPY2 for X68000 Version *.** Copyright 1989 SHARP/Hudson

送り側　退避
受け側
実行
終了

項目を指定してください
↑↓（選択）↵（確定）ESC（終了）
```

「機能」にカーソルがあることを確認して、[↵]を押します。
次のような メニューが表示されます。

```
COPY2  for  X68000  Version *.** Copyright 1989 SHARP/Hudson

機能    退避
送り側
受け側
実行
終了

退避
復帰

機能を指定してください
[↓][↑]（選択）[↵]（確定）[ESC]（前に戻る）
```

ここで、

退避：ハードディスクのファイルをフロッピーディスクに退避する

復帰：フロッピーディスクからハードディスクに復帰する

から選びます。

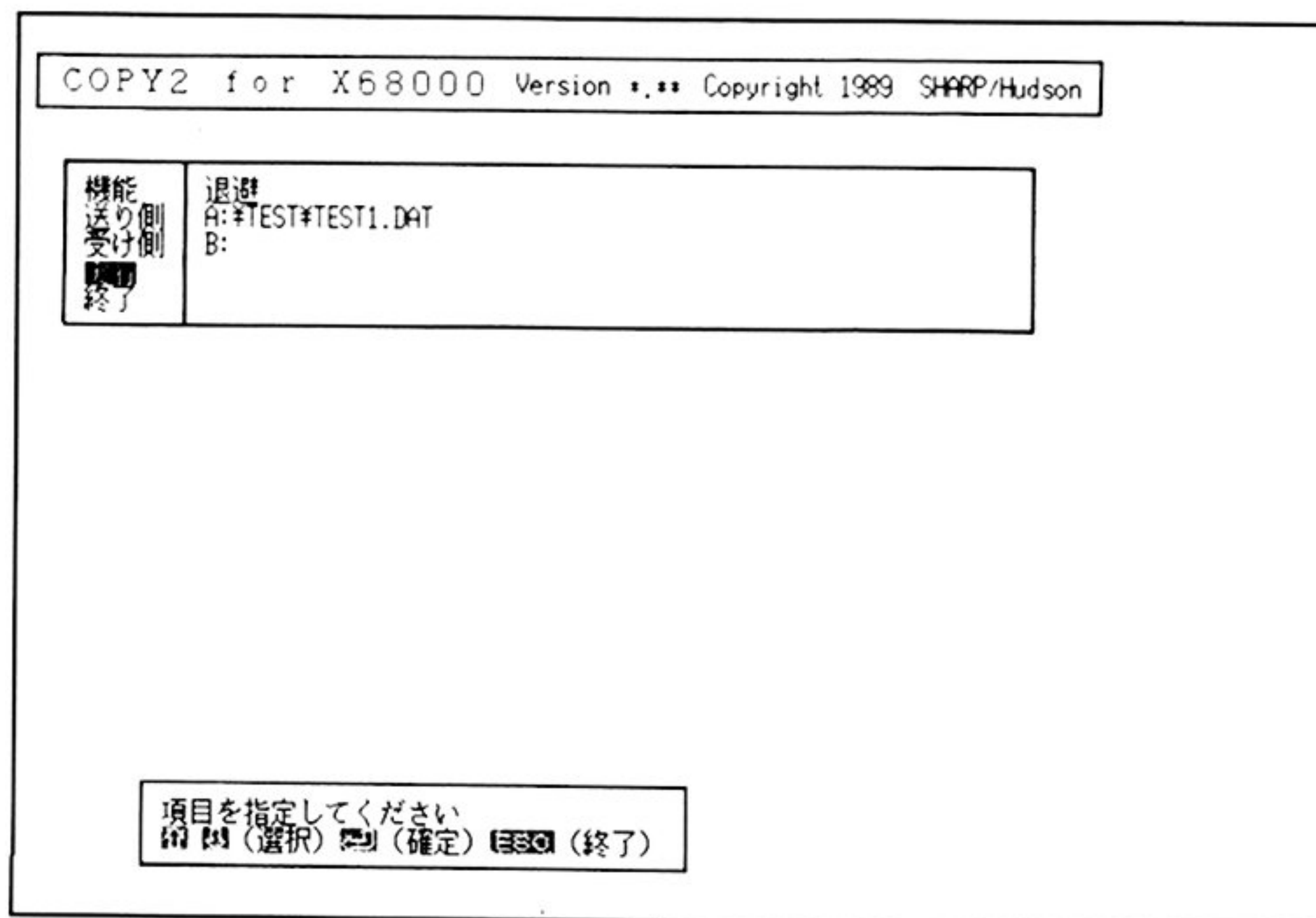
退避するときは、受け側のドライブにフォーマット済みのフロッピーディスク

復帰するときは、送り側のドライブに退避ファイルのあるフロッピーディスク

を挿入し、カーソルを「退避」または「復帰」に合わせ、[↵]を押してください。

次に、「送り側」にカーソルを合わせ[↵]を押すと、送り側のドライブ名を選択するメニューが表示されますので、ドライブ名を[↓][↑]で選び[↵]を押してください。[↵]が押されると、ディレクトリやファイル名が表示されます。退避（復帰）するファイルを[↓][↑]で選び、[↵]を押してください。指定したディレクトリやファイルが画面に表示されます。

こんどは、「受け側」にカーソルを合わせ[↵]を押すと、受け側のドライブ名を選択するメニューが表示されますので、受け側のドライブ名を[↓][↑]で選び[↵]を押してください。「退避」を選んでいるときには、指定したドライブ名が画面に表示されます。また、「復帰」を選んでいるときには、ディレクトリが表示されますので、復帰するディレクトリを[↓][↑]で選び[↵]を押してください。指定したドライブやディレクトリが表示されますので、退避（復帰）する内容を確認してください。

```
COPY2 for X68000 Version *.** Copyright 1989 SHARP/Hudson

機能     退避
送り側   A:¥TEST¥TEST1.DAT
受け側   B:
実行
終了

項目を指定してください
↑↓（選択）↵（確定）ESC（終了）
```

カーソルを「実行」に合わせ、↵を押してください。退避（復帰）が始まります。

1枚のフロッピーディスクで処理が終わらないときは、2枚目、3枚目をセットするようにメッセージが表示されるので、それにしたがってください。特に、復帰するときは退避した順にフロッピーディスクを取りかえてください。

2．コマンド行からの実行

コマンド行で 書式 に従ったパラメータを設定した場合、メニュー画面に入らずにコマンドを実行します。

ハードディスクからフロッピーディスクへファイルを退避するときには、書式(1)を使用します。

書式(1)では、〔<hd：>〕〔<パス名>〕<ファイル名>で指定されたハードディスクのファイルを〔<fd：>〕で指定されたドライブ上のフロッピーディスクに退避します。このとき、1枚のフロッピーディスクに退避しきれないと、2枚目、3枚目を挿入するようメッセージが表示されます。指示に従ってフロッピーディスクを取りかえてください。

逆に、フロッピーディスクからハードディスクへファイルを復帰するときには、書式(2)を使用します。

書式(2)では、〔<fd：>〕<ファイル名>で指定されたフロッピーディスクのファイルを〔<hd：>〕〔<パス名>〕で指定されたハードディスクのディレクトリに復帰します。このとき、複数枚に分けて退避されていたファイルを復帰する場合は、退避した順にフロッピーディスクを取りかえなくてはなりません。

いずれの場合も、ドライブの指定が省略されたときは、カレントドライブが対象になります。パス名が省略されたときは、カレントディレクトリが対象になります。

なお、ファイルを退避した場合、そのファイルはCOPY 2コマンド以外の通常のコマンドで扱うことはできません。

次に退避と復帰の２例をあげます。

1．退避

COPY2　A：¥TEST¥TEST1.DAT　B：⏎

と実行すると、次のメッセージが表示されます。

1番目のディスクをドライブ B:にセットして、何かキーを押してください

メッセージ通りに作業し、任意のキーを押します。もし１枚に収まりきらないときは、次のメッセージが表示されます。

2番目のディスクをドライブ B:にセットして、何かキーを押してください

同様に作業します。次のメッセージが表示されコマンドの実行が終了します。

ファイルの退避が終了しました

この例では、ドライブAのハードディスクのディレクトリTESTにあるTEST1.DATというファイルを、ドライブB上で２枚のフロッピーディスクに退避しました。

2．復帰

COPY2　B：TEST1.DAT　A：¥TEST　／R ⏎

と実行すると、次のメッセージが表示されます。

1番目のディスクをドライブ B：にセットして、何かキーを押してください

メッセージ通りに作業し、任意のキーを押します。

2番目のディスクをドライブ B：にセットして、何かキーを押してください

同様に作業します。次のメッセージが表示されコマンドの実行が終了します。

ファイルの復帰が終了しました

この例では、1.で退避したハードディスクのTEST1.DATというファイルを、ドライブAのディレクトリTESTに復帰しました。

COPYALL

外部コマンド

機　能

ファイルおよびディレクトリのコピー

書　式

COPYALL 〔／D〕〔／T|／N〕〔／U〕〔／V〕〔／S〕〔〈D1：〉〕〔〈パス名1〉〕〈ファイル名1〉 {〈d2：〉 | 〈パス名2〉}

解　説

〔〈d1：〉〕〔〈パス名1〉〕〈ファイル名1〉で指定したファイルを{〈d2：〉|〈パス名2〉}で指定されたパス以下にコピーします。このとき、転送先のディスクがいっぱいになった場合、ディスクを交換するようにメッセージを表示し、交換すると引き続きコピーを行います。

COPYALLコマンドは種別の違うディスクのオールコピーもできますし、COPYコマンドと異なり、新しいファイルだけをコピーするといった選択的なコピーやサブディレクトリの内容までコピーできます。また、BACKUPコマンドがシステム全体のバックアップを作成するために使用するのに対し、COPYALLコマンドは新しく作ったファイルだけをバックアップするときなどに利用できます。

COPYALLコマンドでは、以下のスイッチが使用できます。

・／D

サブディレクトリまでコピーしない場合に指定します。このスイッチがなければ、サブディレクトリ以下をコピー先にも同じサブディレクトリを作りながらコピーします。

・／T

コピー元とコピー先のファイルを比較し、コピー元のファイルの日付が新しいかコピー先に存在しないファイルであればコピーします。このスイッチがなければすべてコピーします。

・／N

コピー元とコピー先のファイルを比較し、コピー元のファイルの日付が新しければコピーします。コピー先に存在しないファイルはコピーしません。

・／U

指定したファイル名が小文字のとき、大文字にしてコピーします。

・／V

実際にはコピーせず、コピー作業の内容を表示します。ワイルドカードを使って不特定のファイルをコピーするときに、コピーするファイル名を確認するときなどに使います。

・／S

システムファイル（HUMAN.SYS）を除いてコピーします。

これらのスイッチのうち、／Tと／Nを同時に指定することはできません。

●ドライブA（ハードディスク）のすべてのファイルをドライブB（フロッピーディスク）にコピーする

COPYALL A：¥＊.＊ B：⏎

●ドライブAのルートディレクトリのファイルだけをドライブBにコピーする

COPYALL ／D A：¥＊.＊ B：⏎

●ドライブAの日付の新しくなったファイルおよびドライブBに存在しないファイルをドライブBにコピーする

COPYALL ／T A：¥＊.＊ B：⏎

●ドライブAのルートディレクトリで日付の新しくなったファイルおよびドライブBに存在しないファイルをドライブBにコピーする

COPYALL ／D ／T A：¥＊.＊ B：⏎

●ドライブBのファイルと比較して、Aドライブの日付の新しいファイルだけをコピーする

COPYALL ／N A：¥＊.＊ B：⏎

●ドライブAのルートディレクトリにあるファイルのうち、先頭にtest（またはTEST）があるファイルを、すべて大文字にしてドライブBにコピーする

COPYALL ／U A：¥TEST＊.＊ B：⏎

●ドライブAのルートディレクトリTESTにある日付の新しいファイルだけをドライブBにコピーするとき、コピーするファイル名を確認する

COPYALL ／V ／N A：¥TEST¥＊.＊ B：⏎

●ドライブAのルートディレクトリのシステムファイル（HUMAN.SYS）を除いたファイルすべてをドライブBにコピーする

COPYALL ／S A：¥＊.＊ B：⏎

CTTY

内部コマンド

機能　入出力デバイスの変更

書式　CTTY　〈デバイス名〉

解説

CTTYコマンドは、入出力デバイスの変更を行います。標準の入出力デバイス(キーボードおよびディスプレイ)と補助入出力機器(RS-232C回線)を切りかえて入出力を行うときに使います。

〈デバイス名〉には、入出力の可能なデバイス名を指定します。

```
CTTY  AUX ↵
```

とすると、コマンドの入出力をキーボード、ディスプレイから補助入出力機器(RS-232C回線)に変更します。

もとの入出力機器に戻すときは、補助入出力機器側から、

```
CTTY  CON ↵
```

と入力します。これでコマンドの入出力は補助入出力機器からキーボード、ディスプレイに復帰します。

DATE

内部コマンド

機　能　日付の表示・設定

書　式　DATE 〔〈yyyy〉－〈mm〉－〈dd〉〕

解　説

Human68k は、クロック(時計)により、日付・時刻を管理しています(電源を切っても常に日付と時刻は保持されています)。これらの時間情報は、ファイルを作成・更新したときにディレクトリに記録されます。

DATE コマンドは、これらの時間情報のうち〝日付〟を表示したり、設定したりするものです。

DATE ↵

と入力すると、次のように表示されます。

現在の日付は〈yyyy〉－〈mm〉－〈dd〉(〈w〉)です
日付を入力してください：

ここで、〈yyyy〉は西暦による年、〈mm〉は月、〈dd〉は日、〈w〉は曜日(1文字)を表します。

ここに表示されている日付を変更しないときは、そのままリターンキーを押してください。

変更する場合は、次に示すような範囲内の数字を、それぞれハイフン(－)またはスラッシュ(／)で区切って入力します。年、月、日のいずれも、省略することはできません。

年：〈yyyy〉　＝1980～2079、または0～99
月：〈mm〉　＝1～12
日：〈dd〉　＝1～31

曜日、月の大小、うるう年については、システムが自動的に管理しますので、意識する必要はありません。

なお、年について2桁以下で入力する場合は、0～79に対しては2000～2079が、80～99に対しては1980～1999が、それぞれ対応します。たとえば、

95-11-7 ↵

と入力すれば、1995年11月7日を指定したことになります。また、

1-1-1 ↵

と入力すれば、2001年1月1日を指定したことになります。

入力した日付が誤っている場合、次のようにエラーメッセージが表示されますので、

今度は正しい範囲内の数値を入力してください。

日付の指定が違います
日付を入力してください：

DATEコマンドの後ろに、直接日付を指定して入力することも可能です。

DATE　95-11-7 ⏎

このときは、日付が正しければメッセージは何も表示されず、そのまま日付の更新がなされます。日付が誤っていると、前述と同じエラーメッセージが表示されますので、再入力してください。

DEL(ERASE)

内部コマンド

機　能　ファイルの削除

書　式　DEL 〔〈d：〉〕〔〈パス名〉〕〈ファイル名〉〔／Y｜／Q〕

別表記　ERASE

解　説

DEL コマンドは、ファイルを削除するときに使います。

誤ってファイル名を指定してコマンドを実行してしまうと、ファイルを回復することは不可能となります。あらかじめ DIR コマンドなどでファイル名を確認してから実行してください。

```
DEL  MSTFL.100 ↵
```

では、ファイル MSTFL．100が削除されます。このとき、削除されたファイル名(ここでは MSTFL．100)が表示されます。

```
DEL  MSTFL.* ↵
```

とすると、ファイル名に MSTFL を持ったすべてのファイル(拡張子は何でもよい)が削除されます。このとき、削除されたすべてのファイル名が画面に表示されます。

```
DEL  *.* ↵
```

では、カレントディレクトリ中のすべてのファイルが、また、

```
DEL  〈パス名〉 ↵
```

では、〈パス名〉で指定されたディレクトリ中のすべてのファイルが、それぞれ削除されます。このとき、削除されたすべてのファイル名が画面に表示されます。
なお、削除の対象となるファイルが複数である場合は、次の確認メッセージが表示されます。

よろしいですか 〈Y／N〉

削除してもよければ[Y]、そうでなければ[N]を押してください。ただし、スイッチ／Y をつけると確認のメッセージが表示されないので、バッチファイルで用いるときに便利です。また、スイッチ／Qをつけると削除の対象となるファイル名とともに確認のメッセージが表示されたのち、削除を行います。これを用いると複数のファイルを確認しながら削除できます。

注意

ファイルを削除するとき、ドライブ名やディレクトリ名が、仮想ドライブや仮想ディレクトリになっているときは、次のことに注意してください。

削除しようとするファイルのディレクトリに、仮想ドライブが指定されているときは、削除を指定した実体のファイルのほかに、仮想ドライブの中にある同名のファイルも削除されてしまいます。たとえば、仮想ドライブの指定が、

C：＝ A：¥TEST

のとき、ドライブAとドライブCに次のようなファイルがあるときには、

コマンド行で

DEL　A:¥TEST¥SAMPLE1 .DAT ⏎

と、DELコマンドを実行すると、A:¥TEST上の実体のファイルだけでなく、仮想ドライブC上のファイルも削除されてしまいます。

また、逆に削除しようとするファイルが仮想ドライブや仮想ディレクトリにあるときは、実体としてあるファイルも削除されてしまいます。たとえば、上と同じような設定の場合、コマンド行で

DEL　C：SAMPLE1 .DAT ⏎

と、DELコマンドを実行すると、実際にDELコマンドが行う作業は、

DEL　A：¥TEST¥SAMPLE1 .DAT ⏎

となり、結果的には仮想ドライブC上のSAMPLE1 .DATも削除されてしまいます。

このように、削除しようとして指定したファイルが、実体のドライブやディレクトリ上のファイルだけでなく、仮想ドライブやディレクトリ上のファイルも削除されてしまいますので、ご注意ください（仮想ドライブや仮想ディレクトリについては、SUBSTコマンドをご覧ください)。

DIR

内部コマンド

機　能　ディレクトリ(ファイル名一覧)の表示

書　式　DIR 〔<d：>〕〔<パス名>〕〔<ファイル名>〕〔／N〕〔／L〕〔／T〕〔／R〕〔／P〕〔／W〕

解　説　DIRコマンドは、ファイルおよびサブディレクトリの一覧("ディレクトリ情報"または単に"ディレクトリ"と呼びます)を表示します。このとき表示される情報は、次のとおりです。

ディスクのボリュームラベル名
カレントドライブ名およびカレントディレクトリ名
表示されるファイルおよびサブディレクトリの合計数「nnnファイル」
現在使用中の合計容量「nnnK Byte 使用中」
ディスクの残り容量「nnnK Byte 使用可能」
ディレクトリ内のファイルだけの容量「nnnK Byte 使用」

ファイル名またはサブディレクトリ名
ファイルの大きさ(バイト単位)
ファイルが最後に更新された日付
ファイルが最後に更新された時刻

パラメータを省略すると、カレントドライブのディレクトリ情報を表示します。

DIR ⏎

ドライブ名のみ指定すると、そのドライブのディレクトリ情報を表示します。

DIR　B： ⏎

パス名を指定すると、指定したサブディレクトリのディレクトリ情報を表示します。

DIR　¥住所 ⏎

ファイル名を指定すると、指定したファイルについてだけのディレクトリ情報を表示します。

DIR　社員名簿 ⏎

この場合、ファイル名としてワイルドカード(*、?)を使用することができます。特定のファイルのみを表示する場合などに便利です。詳細は、「3．3ワイルドカード」を参照してください。

スイッチ／N、／L、／T、／Rはファイルを次のようにソート(並べかえ)して表示します。

／N…名前でソート（昇順）
／L…大きさ（バイト単位）でソート（昇順）
／T…時刻でソート（昇順）
／R…上記の3つのスイッチとともにつけると逆順にソート

また、／N、／L、／Tは同時に指定でき、先に指定されたものから優先的にソートします。たとえば、／T／Lと指定した場合、同じ時刻のものは大きさでソートします。

スイッチ／Pを指定すると、1画面分の表示を行って停止します。再開するときは任意のキーを押します。

スイッチ／Wを指定すると、1行に4つずつ(横64文字モードの場合は2つずつ)ファイル名およびサブディレクトリ名のみが表示されます。ファイルに付随する情報は表示されません。

DIRの実行例です。

```
 Human68k              A:¥
    10 ファイル       102K Byte 使用中       1119K Byte 使用可能
 ファイル使用量         98K Byte 使用
FORM                   <dir>     95-07-07  14:45:00
MAGAZINE               <dir>     95-07-08  15:08:00
EWS                    <dir>     95-09-04  13:28:00
X68K                   <dir>     95-11-03  19:07:00
DOC             TMP         2115  95-12-09  19:11:00
MANA            DOC        35110  95-01-26  12:00:00
JUN             DOC        37519  95-04-26  12:18:00
ETU             DOC         4863  95-09-20   8:09:00
SANAE           DOC         5219  95-12-07  19:30:00
KYOKO           DOC        12134  95-12-28  19:55:00
```

注意

ディレクトリ情報を表示するときに、指定したディレクトリに、仮想ドライブや仮想ディレクトリが含まれているときは、次のことに注意してください。

DIRコマンドを使って表示される情報のうち、「使用中」、「使用」、「使用可能」に表示されるディスクの使用量は、指定したドライブが仮想ドライブのときは、その実体となるドライブの使用量となります。たとえば、ドライブAが内蔵ハードディスクの場合に、仮想ドライブの指定が、

C：=A：¥TEST

と指定されているとき、

DIR　C：↵

とコマンド行で入力すると、

DIR　A：¥TEST　↵

と同じディレクトリ情報が表示されます。

つまり、ドライブCはハードディスクではないにもかかわらずハードディスクの使用量が表示されてしまいます。

このように、仮想ドライブや仮想ディレクトリを含むディレクトリを指定すると、実体となるドライブのディスクの使用量が表示されます。したがって、ディスクの使用量よりファイルの使用量が大きいときや、逆に小さいときがあります。このような状態では、使用可能量より大きなファイルが作成できたり、逆に使用可能量に満たないにもかかわらずファイルが作成できないなどということが起きますので、ご注意ください（仮想ドライブや仮想ディレクトリについては、SUBSTコマンドをご覧ください）。

DISKCOPY

外部コマンド

機　能	ディスク全体のコピー・比較照合
書　式	DISKCOPY 〔<d 1 :>〕〔<d 2 :>〕〔/V〕
解　説	DISKCOPYコマンドは、あるディスクの内容を、他のディスクにすべてコピーします。

このコマンドでコピーされたディスクは、コピー元のディスクとまったく同一の状態となりますから、バックアップコピーを作成するときに使うことができます。

また、DISKCOPYコマンドは2つのドライブにあるディスクの内容を比較照合することができます。

1．メニュー画面からの実行

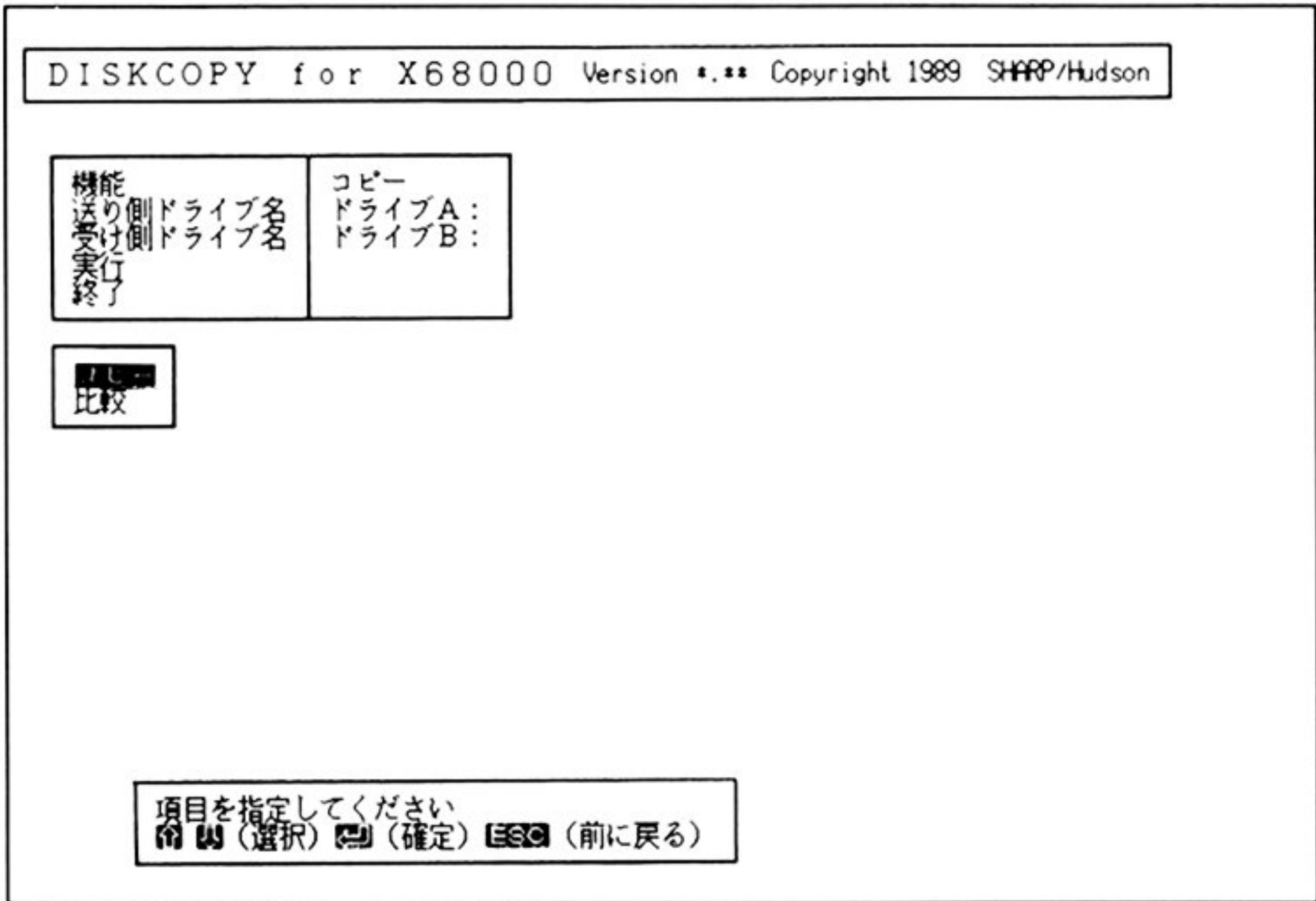
DISKCOPY ↵

と入力すると画面は次のようなメニュー画面になります。

```
DISKCOPY for X68000 Version *.** Copyright 1989 SHARP/Hudson

機能              コピー
送り側ドライブ名  ドライブA：
受け側ドライブ名  ドライブB：
実行
終了

項目を指定してください
↑↓（選択）↵（確定）ESC（終了）
```

「機能」にカーソルがあることを確認して、↵を押します。

次のようなメニューが表示されます。

```
DISKCOPY for X68000 Version *.** Copyright 1989 SHARP/Hudson

機能               コピー
送り側ドライブ名   ドライブA：
受け側ドライブ名   ドライブB：
実行
終了

コピー
比較

項目を指定してください
↑ ↓（選択）↵（確定）ESC（前に戻る）
```

ここで、

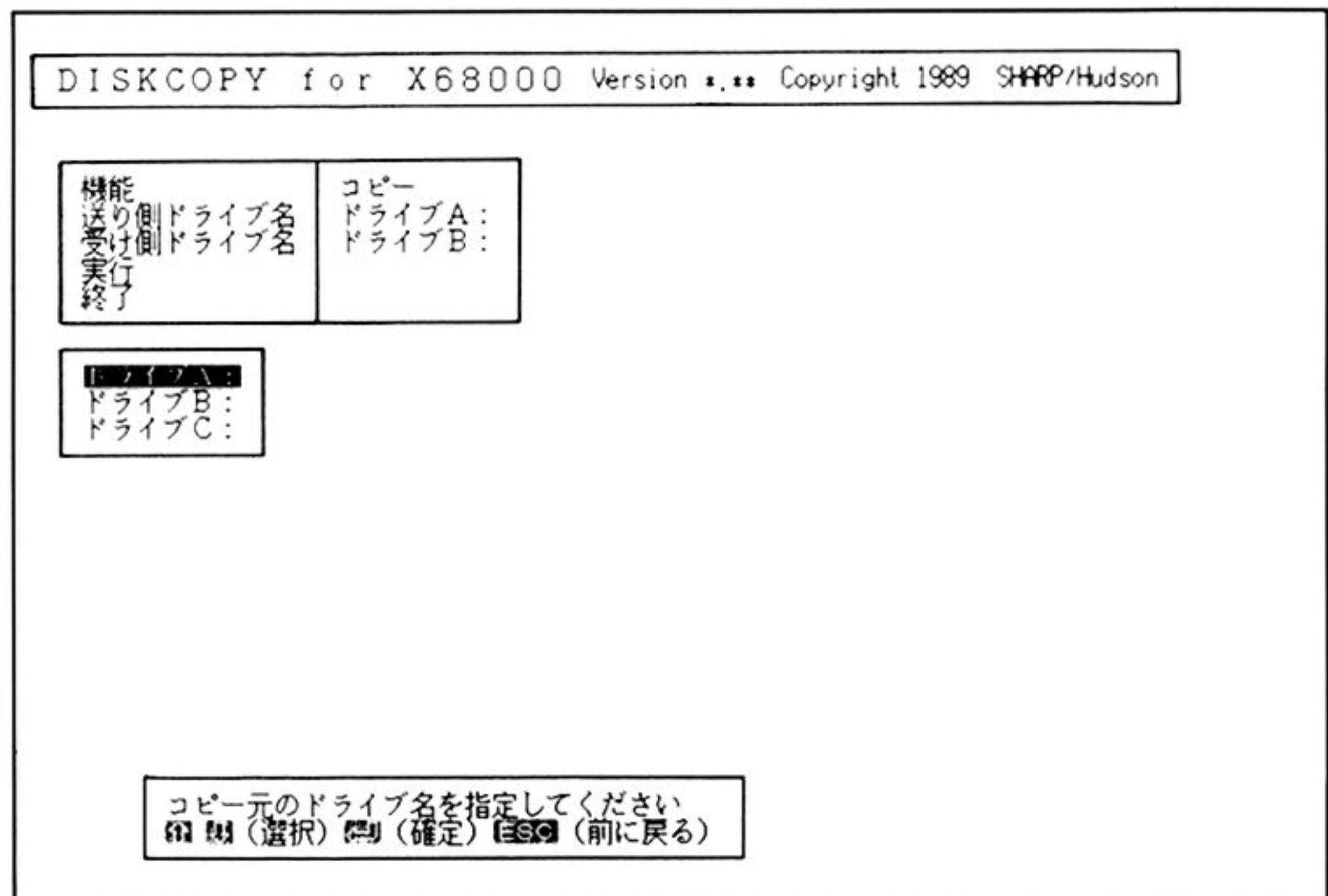
　コピー：ディスク全体の照合

　比較　：ディスク全体の比較照合

から選びます。(「機能」に現在表示されている機能を使うときには、この操作は必要ありません。)

カーソルを「コピー」または「比較」に合わせ、↵を押してください。

次に、「送り側ドライブ名」にカーソルを合わせ、↵を押すと、次のようなメニューが表示されます。

```
DISKCOPY for X68000 Version *.** Copyright 1989 SHARP/Hudson

機能               コピー
送り側ドライブ名   ドライブA：
受け側ドライブ名   ドライブB：
実行
終了

ドライブA：
ドライブB：
ドライブC：

コピー元のドライブ名を指定してください
↑ ↓（選択）↵（確定）ESC（前に戻る）
```

コピーするときは、原本となるディスクのドライブ名を、

比較するときは、比較するディスクのドライブ名を、

選びます。

同じように、「受け側ドライブ名」も

コピーするときは、フォーマット済みのフロッピーディスクを挿入するドライブ名を、

比較するときは、比較するもう一方のディスクのドライブ名を、

選びます。

ここで、送り側と受け側のドライブにフロッピーディスクを挿入し、カーソルを「実行」に合わせ、[↵]を押してください。ディスク全体のコピー・比較照合が始まります。

コピーが終ると、

コピーは終了しました　何かキーを押してください

比較が終ると、

比較したディスクの内容が一致しています
何かキーを押してください

または

比較したディスクの内容は異なっています　他のディスクを比較しますか？
[Y]（継続）[N]（前に戻る）[ESC]（前に戻る）

と表示されます。指示にしたがってキーを押してください。

なお、DISKCOPY コマンドは次の場合、実行できないので注意してください。

・同一ドライブでのコピーと比較照合
・コピー元のディスクとコピー先のディスクの種別が異なるとき
・コピー先のディスクにライトプロテクトがかかっているとき
・コピー先のディスクがフォーマットされていないとき

DISKCOPY コマンドはハードディスクに対しても使用することができます。ただし、コピー元のハードディスクの容量よりコピー先のハードディスクの容量が少ないときは実行できません。また、比較照合の場合は、同じ容量でないと実行できません。

2．コマンド行からの実行

コマンド行で [書式] に従ったパラメータを設定した場合、メニュー画面に入らずにコマンドを実行します。

1．ディスクコピー

<d 1：>で指定されたドライブ上のディスク(コピー元)の全内容を<d 2：>で指定されたドライブ上のディスク(コピー先)にコピーします。

コピー先となるディスクは、Human68k フォーマットでフォーマット(初期化)されていなくてはなりません。もし、新しいディスクを使用する際には、前もって FORMAT コマンドでフォーマットしておいてください。

DISKCOPY　A：　B： [↵]

と入力すると、次のようなメッセージが表示されます。

ドライブ A からドライブ B へコピーします。何かキーを押してください

ドライブ A にコピー元のディスクを、ドライブ B にコピー先のディスクをそれぞれ入れ、任意のキーを押すとコピーが始まります。コピー中は、次のメッセージが表示されます。

コピー中です…
ディスクコピー終了しました
他のディスクをコピーしますか？<Y／N>

ここで、続けてコピーを行いたければ[Y]を、終了したければ[N]を押してください。
[Y]を押したときは、自動的にフロッピーディスクのみがイジェクトされます。
なお、次のように同一のドライブを指定すると、

DISKCOPY　A：　A： [↵]

次のようなエラーメッセージが表示され、コピーは行われずにコマンドが終了します。

同一ドライブでのコピーはできません

また、コピー元のディスクとコピー先のディスクの種別が違っていると、次のようなエラーメッセージが表示され、コマンドは終了します。

種別の違うドライブなので、コピーはできません

2．ディスクの比較照合

スイッチ／V を指定して DISKCOPY コマンドを実行すると、ドライブ<d 1：>上のディスクの内容とドライブ<d 2：>上のディスクの内容の比較照合を行います。

DISKCOPY　A：　B：　／V ↵

と入力すると、次のようなメッセージが表示されます。

ドライブＡとドライブＢの内容を比較します。何かキーを押してください

指示されたとおりにディスクを入れ、任意のキーを押すと比較が始まります。比較中は、次のようなメッセージが表示されます。

比較中です・・・

比較が終り、２つのディスクの内容がまったく同一であったときに次のように表示されます。

比較を終了しました。比較したディスクの内容は一致しています
他のディスクを比較しますか？<Y／N>

ここで、続けて比較を行いたければ[Y]を、終了したければ[N]を押してください。
比較の結果、２つのディスク間で内容の異なるセクタが発見されたときには、次のようなメッセージが表示されます。

比較したディスクの内容は異っています
他のディスクを比較しますか？<Y／N>

ここで、続けて比較を行いたければ[Y]を、終了したければ[N]を押してください。[Y]を押したときは、自動的にフロッピーディスクのみイジェクトされます。
なお、次のように同一のドライブを指定すると、

DISKCOPY　A：　A：　／V ↵

次のようなエラーメッセージが表示され、比較は行われずにコマンドが終了します。

同一ドライブでの比較はできません

また、２つのディスクの種別が違っていると、次のようなエラーメッセージが表示され、コマンドは終了します。

種別の違うディスクなので、比較はできません

３．ハードディスク間のコピー・比較照合

DISKCOPYコマンドは、ハードディスクに対しても使用することができます。ただし、コピー元のハードディスクの容量よりコピー先のハードディスクの容量が少ないときには、次のようなエラーメッセージを表示し、コマンドを終了します。

種別の違うディスクなので、コピーできません

比較照合の場合も、双方のハードディスクの容量が異なるときには、次のようなエラーメッセージを表示し、コマンドを終了します。

種別の違うディスクなので、比較できません

注意

●ディスクコピー時に、ドライブの指定がされていないと、次のようなメッセージが表示されますので、ドライブ名を入力してください。

コピー元のドライブを指定してください：
コピー先のドライブを指定してください：

●比較照合時に、ドライブの指定がされていないと、次のようなメッセージが表示されますので、ドライブ名を入力してください。

比較元のドライブを指定してください：
比較先のドライブを指定してください：

●次の場合はコピーができません。それぞれに適切な処置をしてから、再度コマンドを実行してください。

1．コピー先のディスクにライトプロテクトがかかっているとき
2．コピー先のディスクがフォーマットされていないとき

DRIVE

外部コマンド

機能　ドライブの種類、状態の表示・ドライブ名の交換

書式　DRIVE 〔/V〕〔/D〕〔‹d 1 :›〕〔‹d 2 :›〕

解説　DRIVEコマンドは、ドライブ名‹d 1 :›と‹d 2 :›を指定するかどうかで動作が異なります。

‹d 1 :›、‹d 2 :›でのドライブ名の指定がない場合は、全ドライブのタイプを表示します。

DRIVE ⏎

A：ハード　ディスク　　ユニット番号………0
B：2 HD(1 MB)　　ユニット番号………0
C：2 HD(1 MB)　　ユニット番号………1
D：RAM　DISK　　ユニット番号………0
E：仮想ドライブです
|
Z：仮想ドライブです

仮想ドライブについては、SUBSTコマンドを参照してください。

‹d 1 :›のドライブ名だけを指定すると、そのドライブの状態を表示します。

DRIVE　A：⏎

A：ハードディスク　　ユニット番号……0
1セクタあたりのバイト数 ……………………1024
1クラスタあたりのセクタ数 …………………1
総クラスタ数……………………………………20155
ファイルアロケーションの先頭セクタ番号 ……1
ファイルアロケーションのセクタ数 …………40
ルートディレクトリの先頭セクタ番号…………81
ルートディレクトリの最大個数 ………………512
データ領域の先頭セクタ番号 …………………97
アクセス…………………………………………可
イジェクト ………………………………………不可
書き込み…………………………………………可

‹d 1 :›、‹d 2 :›のドライブ名とも指定した場合は、指定したドライブ名を交換します。

DRIVE　A：　D：⏎

ドライブＡ：とドライブＤ：を入れ替えました
カレントＡ：をドライブＤ：に切り替えました
Ａ：RAM　DISK　　　ユニット番号……０
Ｂ：２ HD（１ MB）　　ユニット番号……０
Ｃ：２ HD（１ MB）　　ユニット番号……１
Ｄ：ハード　ディスク　　ユニット番号……０
Ｅ：仮想ドライブです
｜
Ｚ：仮想ドライブです

この例では、DRIVEコマンドの実行前はＡ：ハードディスク、Ｄ：RAMディスクだったのが、実行後はＡ：RAMディスク、Ｄ：ハードディスクとなります。

／V　ボリュームラベルを付けて全ドライブのタイプ表示を行います
／D　すべてのドライブ名を初期状態に戻します

／Vは、ハードディスクの領域を分割して利用しているときに、ボリュームラベルを確認するためのものです。

なお、／Dを付けたときは、ドライブ名を指定しないでください。

DUMP

外部コマンド

機 能

16進表現・文字によるファイル内容の表示

書 式

DUMP 〈ファイル名〉〔/〈開始アドレス〉〔,〈バイト数〉〕〕

解 説

DUMPコマンドは、〈ファイル名〉で指定されたファイルの内容を、バイト単位の16進表現とアスキー文字により表示します。

DUMPコマンドでは、TYPEコマンドでは表示できないようなコントロール文字を含んだバイナリファイルでもコード単位で表示することができます。

〈開始アドレス〉および〈バイト数〉を指定すると、〈開始アドレス〉番地（ファイルの先頭からのオフセット値）から〈バイト数〉分だけダンプします。このとき、これらのそれぞれの指定値の前に"X"をつけると16進、何もつけないと10進による指定となります。

〈バイト数〉を省略し、〈開始アドレス〉だけを指定したときは、指定番地からファイルの最後までダンプされます。

```
DUMP MSTFL.100 [↵]
```

と入力すると、ファイルMSTFL.100の内容が表示されます。このとき、画面の左側部分には、このファイルの最初からのアドレス相対値が、中央部分にはファイル内容のバイト単位の16進表現が、右側部分には該当するアスキー文字が、それぞれ表示されます。右側部分のアスキー文字は、表現不可能な文字についてはピリオド（.）で表示されます。

● DUMPの中断

大きなファイルをダンプすると、1画面に収まり切らず、スクロールしてしまいます。このような場合、[CTRL]+[S]を押すことにより、表示を途中で止めることができます。いったん止めた後は、任意にキーを押せばスクロールは再開します。

● DUMPの終了

ダンプの実行中に[CTRL]+[C]を押せば、処理は中止されて、コマンドモードに戻ります。

EXIT

機　能　子プロセスとして起動されたCOMMAND.Xからの親プロセスへのリターン

書　式　EXIT

解　説

Human68kシステムでは、あるプログラムの内部から子プロセスを生成し、これに実行制御を一時的に移すことができるような機能をもっています。

この子プロセス機能を利用して、COMMAND.Xプログラムを子プロセスとして起動できるようにしたアプリケーションプログラムでは、そのアプリケーションプログラムの途中の状態を保持したまま、Human68kのコマンドモードに移行することができます。すなわち、アプリケーションプログラムを終了させずにHuman68kのコマンドを使うことができるわけです。

EXITコマンドは、このように子プロセスとして起動されたCOMMAND.Xから、親のアプリケーションプログラムに戻るために使用します。

FASTIO

外部コマンド

機能

I／O関連の高速化

書式

FASTIO〔キャッシュバッファ容量〕〔パラメータ〕

解説

FASTIOコマンドは、一度アクセスしたディスクのデータをメインメモリに記憶（キャッシュ）していくことで、ディスクへのアクセス回数を少なくすることができ、データの読み書きが高速になります。

コマンドラインからでもCONFIG.SYSに登録しても実行可能です。

登録すると、ドライブ関係のIOCSコールを変更しますので、FASTIO／FASTSEEK／FASTOPENの各プログラムを解除する場合は、登録したときとは逆の順番で解除する必要があります。また、これらが登録された後には、X68030シリーズ以外の機種ではFDDEVICEが登録できません。

新規登録のみ有効なパラメータ

-b数値	キャッシュバッファ容量（キロバイト単位）を設定します
-p数値	先読み量（セクタ単位）を設定します
-s数値	連続転送バッファ容量（キロバイト単位）を設定します
-k数値	オートロック量（セクタ単位）を設定します

新規登録時でも、常駐時でも設定変更が可能なパラメータ

-d	ディレクトリを優先します
-f	FATを優先します
-l	キャッシュをロックします
-w	遅延書込モードにします
-x	IOCSディスクアクセスをエラーとします

新規登録時でも、常駐時でも設定変更が可能なパラメータで、登録時デフォルト

-t	ライトスルーモードにします
-u	キャッシュロックを解除します
-n	優先順位を指定しません

常駐時のみ使用可能なパラメータ

-c	バッファを消去します
-i	キャッシュ状態を表示します
-v	詳しい状態を表示します
-r	常駐を解除します

なお、バッファは［CTRL］＋［F5］でも消去できます。この場合、FASTIO、FASTOPEN、FASTSEEK の各バッファがすべてクリアされます。

FASTOPEN

外部コマンド

機　能　ファイル名の管理環境設定

書　式　FASTOPEN［ネームバッファ数］［パラメータ］

解　説　FASTOPENコマンドは、Human68kのディレクトリの管理を、メインメモリに確保されたバッファ領域内で行い、ディスクへのアクセス回数を少なくすることで、ファイルのアクセスを高速にします。

コマンドラインからでもCONFIG.SYSに登録しても実行可能です。

登録すると、ドライブ関係のIOCSコールを変更しますので、FASTIO／FASTSEEK／FASTOPENの各プログラムを解除する場合は、登録したときとは逆の順番で解除する必要があります。また、これらが登録された後には、X68030シリーズ以外の機種ではFDDEVICEが登録できません。

新規登録のみ有効なパラメータ

-b数値	ネームバッファ数を設定します
-c	大文字小文字を区別します

新規登録時でも、常駐時でも設定変更が可能なパラメータ

-l	バッファをロックします
-x	IOCSディスクアクセスをエラーとします

新規登録時でも、常駐時でも設定変更が可能なパラメータで、登録時デフォルト

-u	バッファロックを解除します
-y	IOCSディスクアクセスでバッファを消去します

常駐時のみ使用可能なパラメータ

-z	全バッファを初期化します
-i	バッファの状態を表示します
-v	詳細な状態を表示します
-r	常駐を解除します

なお、バッファはCTRL＋F5でも消去できます。この場合、FASTIO、FASTOPEN、FASTSEEKの各バッファがすべてクリアされます。

FASTSEEK

外部コマンド

機　能　デバイスの管理領域サイズの設定

書　式　FASTSEEK〔管理領域サイズ〕〔パラメータ〕

解　説

FASTSEEKコマンドは、Human68kのFATの管理（ファイルのつなぎの情報）をメインメモリに確保されたバッファ領域内で行い、ディスクへのアクセス回数を少なくすることで、データの読み書きを高速にします。

コマンドラインからでもCONFIG.SYSに登録しても実行可能です。

登録すると、ドライブ関係のIOCSコールを変更しますので、FASTIO／FASTSEEK／FASTOPENの各プログラムを解除する場合は、登録したときとは逆の順番で解除する必要があります。また、これらが登録された後には、X68030シリーズ以外の機種ではFDDEVICEが登録できません。

新規登録のみ有効なパラメータ

-b数値	管理領域サイズ（キロバイト単位）を設定します

新規登録時でも、常駐時でも設定変更が可能なパラメータ

-x	IOCSディスクアクセスをエラーとします
-l	バッファをロックします

新規登録時でも、常駐時でも設定変更が可能なパラメータで、登録時デフォルト

-u	ロックを解除します
-y	IOCSディスクアクセスでバッファをクリアします

常駐時のみ使用可能なパラメータ

-c	バッファを消去します
-i	キャッシュ状態を表示します
-v	詳しい状態を表示します
-r	常駐を解除します

なお、バッファは[CTRL]＋[F5]でも消去できます。この場合、FASTIO、FASTOPEN、FASTSEEK の各バッファがすべてクリアされます。

FC

外部コマンド

機　能　ファイル内容の比較

書　式　FC 〈ファイル名〉〈ファイル名2〉〔／A〕〔／B〕〔／W〕〔／C〕〔／n〕〔／T〈タブカラム〉〕〔／L〈最大行数〉〕〔／〈一致行数〉〕

解　説　FCコマンドは、2つのファイルの内容を比較し、異なる部分があれば表示します。ファイルの修正を行った後にその個所を確認するときなどに使用します。

FCコマンドで使用できるスイッチは5種類あります。

／A

アスキー比較の指定。2つのファイルを行単位で比較します（1行を1つのデータブロックとして比較します)。

／B

バイナリ比較の指定。2つのファイルをバイト単位で比較します。

／W

ブランク圧縮。連続したスペース、タブを1つのブランクとして処理します。

／C

大・小文字の無視。テキスト中に現れる大文字、小文字を区別しません。

／n

行番号の表示。内容に行番号を付けて表示します。

／T〈タブカラム〉

タブのスペース化。設定されているタブカラムまでスペースとします。〈タブカラム〉には、2、4、8、16が指定できます。

／L〈最大行数〉

比較する最大行数の指定。〈最大行数〉には、比較する行数の最大値を指定します。

／〈一致行数〉

一致部分の行数。〈一致行数〉には、1～9を指定します。〈一致行数〉のテキストが一致していた場合にのみ、同一の内容として判断します。省略時には3が指定されたものと見なします。

なお、／A、／Bの両方とも指定されていないと、アスキー比較が行われます。ただし、比較するファイルの拡張子が次のどれかであるときは、バイナリ比較が行われます。

X、R、Z、SYS、O、A

したがって、これらのファイルをアスキー比較したい場合には、／Aスイッチを必ず付けてください。

●相違点の報告

以下の形式で出力されます。

＋＋＋＋＋＋〈ファイル名1〉
〈一致している部分の最後の1行〉
・
・
〈相違点〉
・
・
〈再び一致した行〉
＋＋＋＋＋＋〈ファイル名2〉
〈一致している部分の最後の1行〉
・
・
〈相違点〉
・
・
〈再び一致した行〉

／Bスイッチによってバイナリ比較をする場合は、出力報告が次のようになります。

アドレス	ファイル1	ファイル2
xxxxxxxx:	YY	ZZ
・	・	・
・	・	・
・	・	・

ここで、xxxxxxxx はファイルの先頭からのバイト数であり、YY、ZZ は相違のあったデータです。

なお、バイナリ比較では、バイト単位で比較するため、1バイトのずれがあっただけの場合でも、すべてのデータが異なる、という報告となります。

FIND

外部コマンド

機　能　ファイル中の指定文字列の検索

書　式　FIND 〔／V〕〔／C〕〔／N〕〔／L〕〔／F〕〈"文字列"〉〔〈ファイル名〉〕

解　説　FINDコマンドは、指定したファイルから指定した文字列を含んだ行をさがし、それらを全部表示します。

〈ファイル名〉にワイルドカードを用いると、複数のファイルを対象に検索を行うことができます。また、各種スイッチを指定することにより、次のようにいろいろな操作を行うことができます。

／V	指定文字列を含まない行を表示します
／C	指定文字列を含む行数などの検索結果を表示します
／N	出力行の行頭に行番号を付けないで表示します
／L	半角英数字の大文字・小文字を区別しないで検索します
／F	文字列を含む行があるファイルのファイル名を各行の行頭に表示します

これらのスイッチのうち、／Fについては、〈ファイル名〉にワイルドカードを指定したときについてのみ有効です。

スイッチは任意に組み合わせて使用することができますが、／Cを指定した場合、／V、／N、／Fは無視されます。

●文字列を含む行を表示します

```
FIND "東京" WKFL.100 ↵
```

ファイルWKFL.100内より、"東京"という文字列を含んだ行をすべて抜き出し、表示します。

●文字列を含まない行を表示します

```
FIND ／V "神奈川" WKFL.100 ↵
```

ファイルWKFL.100内より、"神奈川"という文字列を含まない行をすべて抜き出し、表示します。

●検索結果を表示します

```
FIND ／C "東京" WKFL.100 ↵
```

ファイルWKFL.100内の、"東京"という文字列を含んだ行数、含まない行数、全体の行数を表示します。

●文字列を含む行に行番号を付けないで表示します

FIND　/N　"港区"　WKFL. 100 ↵

ファイル WKFL. 100内より、"港区"という文字列を含んだ行をすべて抜き出し、それに行番号を付けないで表示します。

●文字列を含まない行に行番号を付けないで表示します

FIND　/V　/N　"港区"　WKFL. 100 ↵

ファイル WKFL. 100内より、"港区"という文字列を含まない行をすべて抜き出し、それに行番号を付けないで表示します。

●文字列を含む行があるファイルのファイル名を各行の行頭に表示します

FIND　/F　"東京"　WKFL. * ↵

主ファイル名に WKFL をもつファイル全部について"東京"という文字列を含んだファイルをさがし、文字列を含んだ行のみ表示します。各行の行頭には、その行があるファイルのファイル名が付加されて、表示されます。

なお、FIND コマンドはフィルタとしても使うことができるようになっており、標準入力からのデータ入力も受け付けることができます。

TYPE　WKFL. 200　|　FIND　"東京"

は、TYPE コマンドによって標準出力に出された WKFL. 200の内容をパイプで標準入力につなぎ、これを FIND コマンドに入力させて処理するように指示するものです。

FORMAT

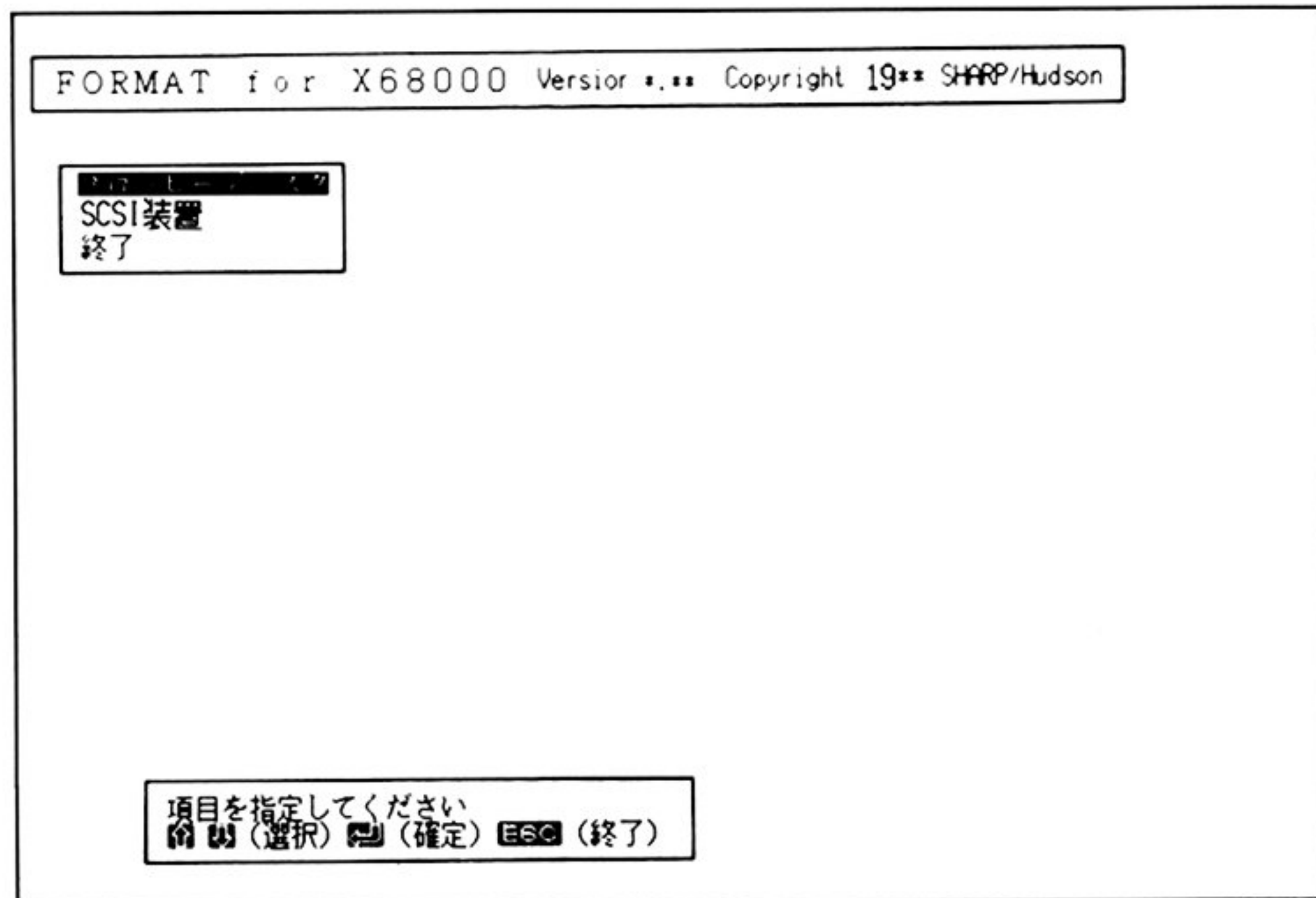
機　能　ディスクのフォーマット(初期化)

書　式　FORMAT　〔<d：>〕〔／S〕〔／C〕〔／V〕

解　説　FORMAT コマンドは、指定したドライブのディスクをフォーマット(初期化)するときに使用します。

1．メニュー画面からの実行

FORMAT ⏎

と入力すると、次のようなメニュー画面が表示されます。

ここでまず、カーソルキーの ↑ ↓ でフロッピーディスクのフォーマットか、SCSI 装置（ハードディスク）のフォーマットかを選択し、⏎ を押します。以下、それぞれのフォーマット方法について説明します。

１．フロッピーディスクのフォーマット

フロッピーディスクのフォーマットを選択すると、画面には次のようなメニュー画面が表示されます。

それぞれの項目には、次のように設定します。

●ドライブ

フォーマットするドライブを指定します。

●ボリューム名

ディスクにつける名前を設定します。

●システム転送

システムの転送を行うかどうかを設定します。システムの転送を行うと、フォーマット終了後にカレントドライブのルートディレクトリにある〝HUMAN.SYS〟をコピーします。

●初期化範囲

フロッピーディスク全体を初期化するのか、すでにフォーマットされているディスクの管理領域（FAT部とディレクトリ部）だけを初期化するのかを設定します。管理領域だけのフォーマットは全体をフォーマットするより高速ですが、一度全体をフォーマットしたフロッピーディスクでなければ使用できません。

それぞれの項目を設定したら、「実行」にカーソルを移動して ↵ を押します。画面には次のようなメッセージが表示されます。

ドライブＢ（２HDディスク）を初期化します
何かキーを押してください　ESC（前に戻る）

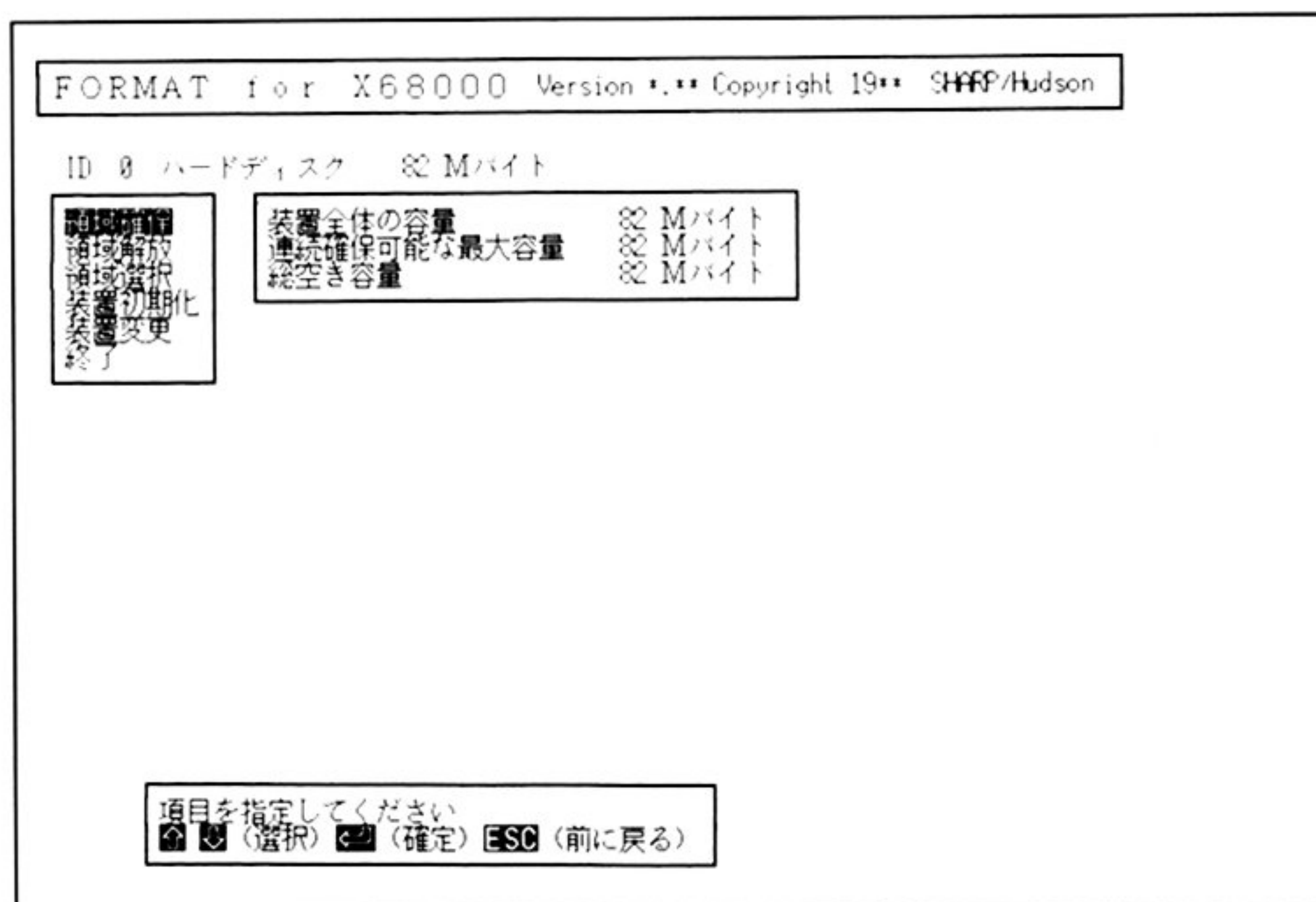
ここで、ドライブd（dは指定したドライブ名）にフロッピーディスクをセットし、任意のキーを押してください。これで、フォーマットが開始されます。フォーマットが終了すると、次のようなメッセージが表示されます。

初期化は終了しました　他のディスクを初期化しますか？
[Y]（継続）[N]（前に戻る）[ESC]（前に戻る）

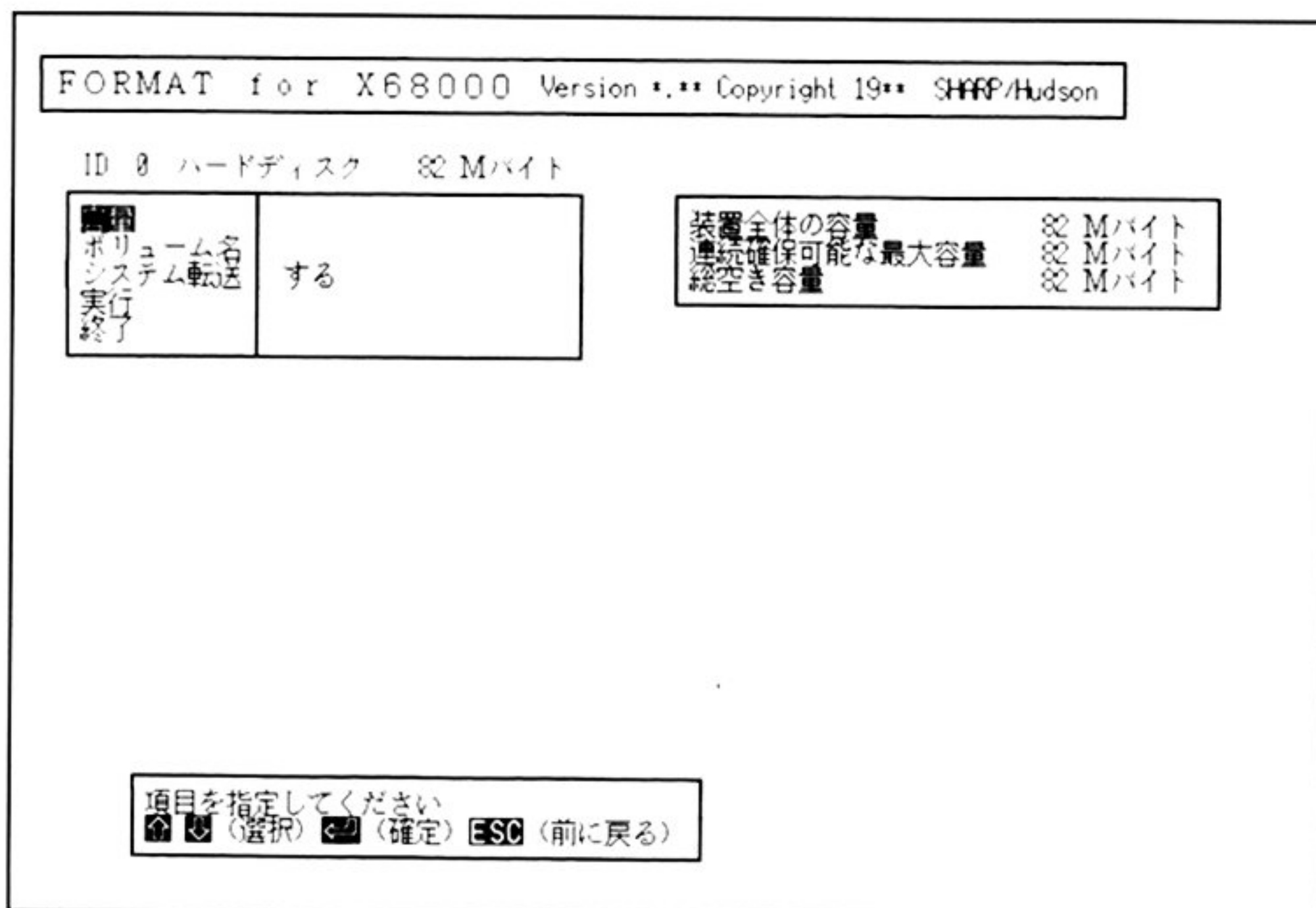
ここで、[Y]を押すと、フロッピーディスクがイジェクトされ、初期化をはじめるメッセージに戻り、続けて別のフロッピーディスクを初期化することができます。[N]または[ESC]を押すと、さきほど項目を設定したメニュー画面に戻ります。

FORMATコマンドを終了したいときは、「終了」にカーソルを移動して [↵] を押してください。

2．ハードディスクのフォーマット

〝SCSI装置〞を選択すると、SCSI装置のメニューが画面に表示されます。フォーマットするSCSI装置を、カーソルキーの[↑][↓]で選択して[↵]を押してください。画面には次のようなメニュー画面が表示されます。

メニュー画面に表示される項目の詳細は、それぞれ次のとおりです。

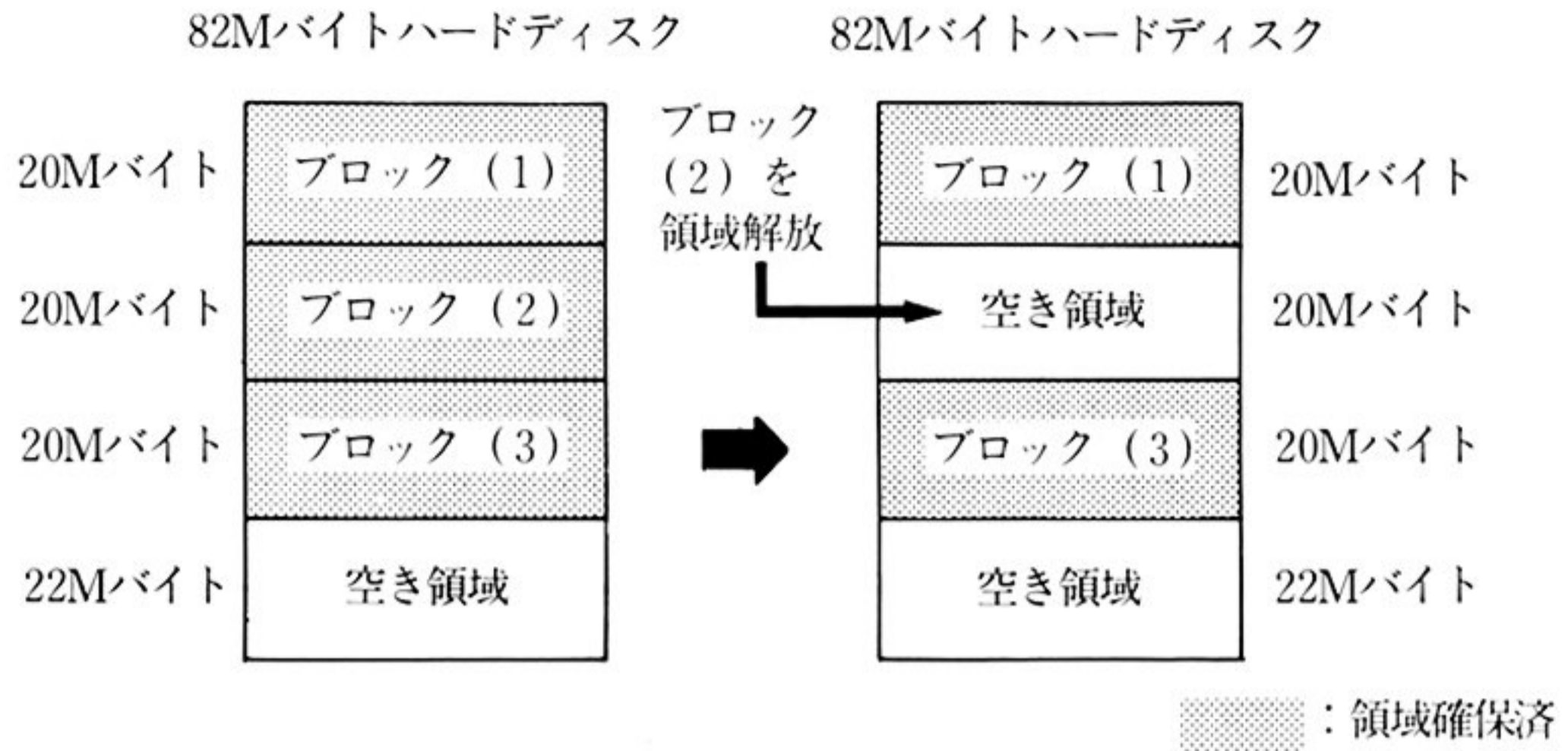
●領域確保

領域の確保を行います。「領域確保」にカーソルを合わせ⏎を押すと、次のようなメニュー画面が表示されますので、次のように設定してください。

・容量

確保する容量を指定します。たとえば、20Mバイト確保するときは "20" と入力します。

なお、すでに領域確保を実行している場合、メニュー画面の右側に表示されている領域の状態を確認してください。ここに表示されている「総空き容量」とは、まだ領域確保していない領域の合計容量、「連続確保可能な最大容量」とは一つの領域（ブロック）が確保できる最大の容量、のことです。

つまり、次のように全体の領域をすでに20Mバイトずつ３ブロック領域確保しているときに、ブロック(2)の領域を領域解放すると、「連続確保可能な最大容量」は20Mバイトの領域と22Mバイトの領域の２つが別々にあるので、「22Mバイト」と表示され、「総空き容量」はこの２つの空き領域の合計の「42Mバイト」と表示されます。

したがって、「総空き容量」は42Mバイトありますが、１つのブロックで42Mバイトを領域確保することはできません。

連続確保可能な最大容量：22Mバイト ------------→ 22Mバイト
総空き容量　　　　　　：22Mバイト ------------→ 42Mバイト

・ボリューム名

確保する領域に付けるボリュームラベルを設定します。

・システム転送

システムの転送を行うかどうかを指定します。システムの転送を行うと、フォーマット終了後にカレントドライブのルートディレクトリにある〝HUMAN.SYS〟と〝COMMAND.X〟をコピーします。

・実行

領域確保を開始します。次のようなメッセージが表示されますので、

領域の確保を行います　よろしいですか？
[Y]（実行）[N]（前に戻る）[ESC]（前に戻る）

ここで、[Y]を押すと領域確保が開始されます。[N]または[ESC]を押すと、元のメニュー画面に戻ります。

領域確保が終了したら、「終了」にカーソルを移動して[⏎]を押します。次のようなメッセージが表示されますので、[Y]を押してください。

ハードディスクの使用には再起動が必要です　リセットしますか？
[Y]（実行）[N]（終了）

・終了

FORMATコマンドを終了します。

●領域解放

確保した領域（ブロック）の解放を行います。解放された領域は再び確保可能な領域となります。「領域解放」にカーソルを合わせ、↵を押すとカーソルがブロックに移動します。ここで解放したいブロックを選び↵を押してください。

このように、2つのブロックが存在する場合、ブロック(1)を解放すると次のようになります。

```
装置全体の容量              82 Mバイト
(2) 自動起動 Human68k       20 Mバイト
連続確保可能な最大容量      42 Mバイト
総空き容量                  62 Mバイト
```

●領域選択

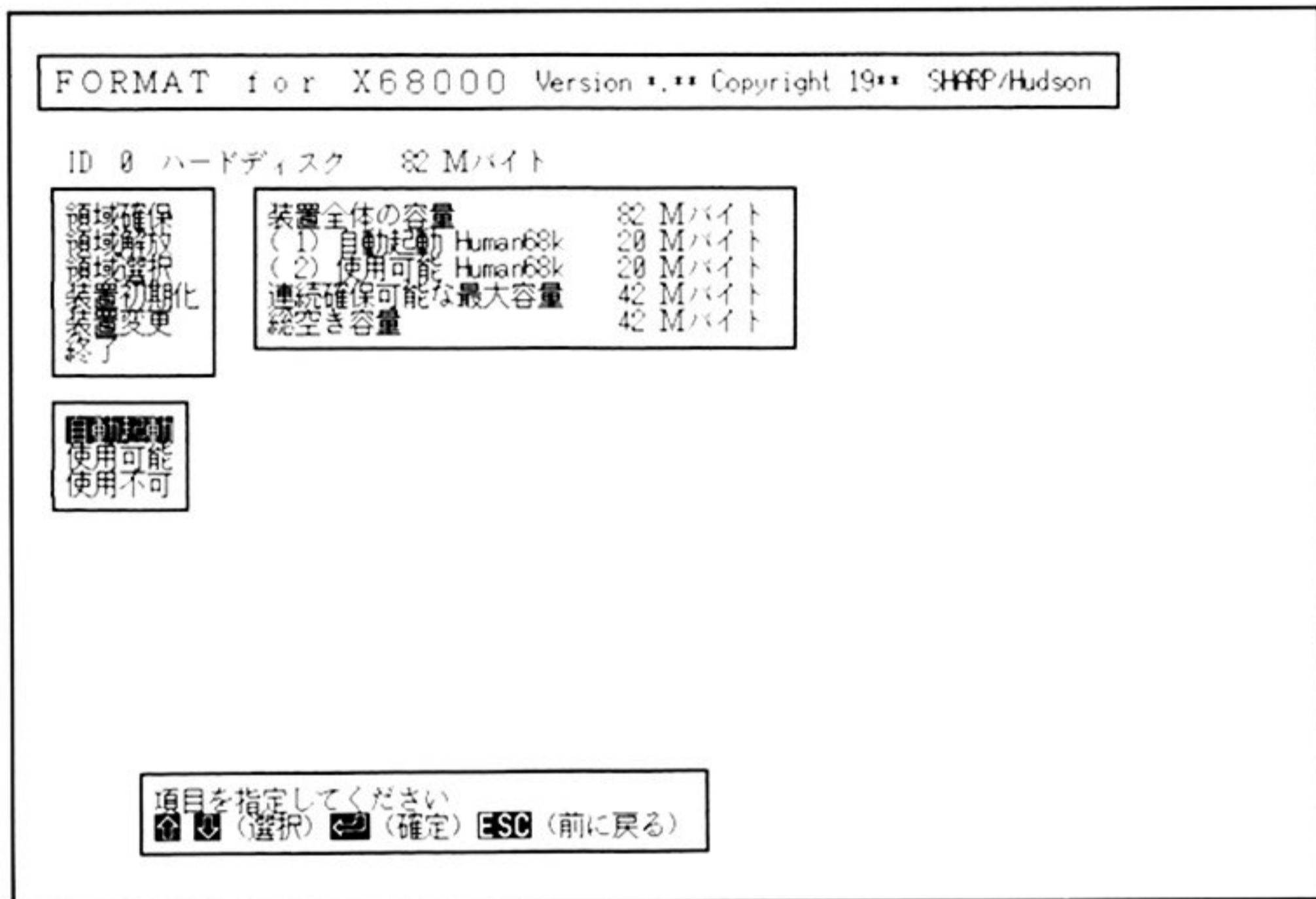
確保された領域（ブロック）の使い方を選びます。使い方には、次の3つがあります。

・自動起動・・・指定された領域から自動的に起動します。
ハードディスクから起動する場合、自動的にシステムを読み込んで実行する領域です。自動起動領域は、1台のハードディスクに必ず1つだけ設定されます。
なお、ここで「自動起動」と設定してもSWITCHコマンドのBOOT（Human68kを起動する装置の設定）でSCSI装置から起動するように設定し、しかもSCSI-ID（接続しているSCSI装置のID番号）の設定を正しく行っていないと、SCSI装置から起動しません。

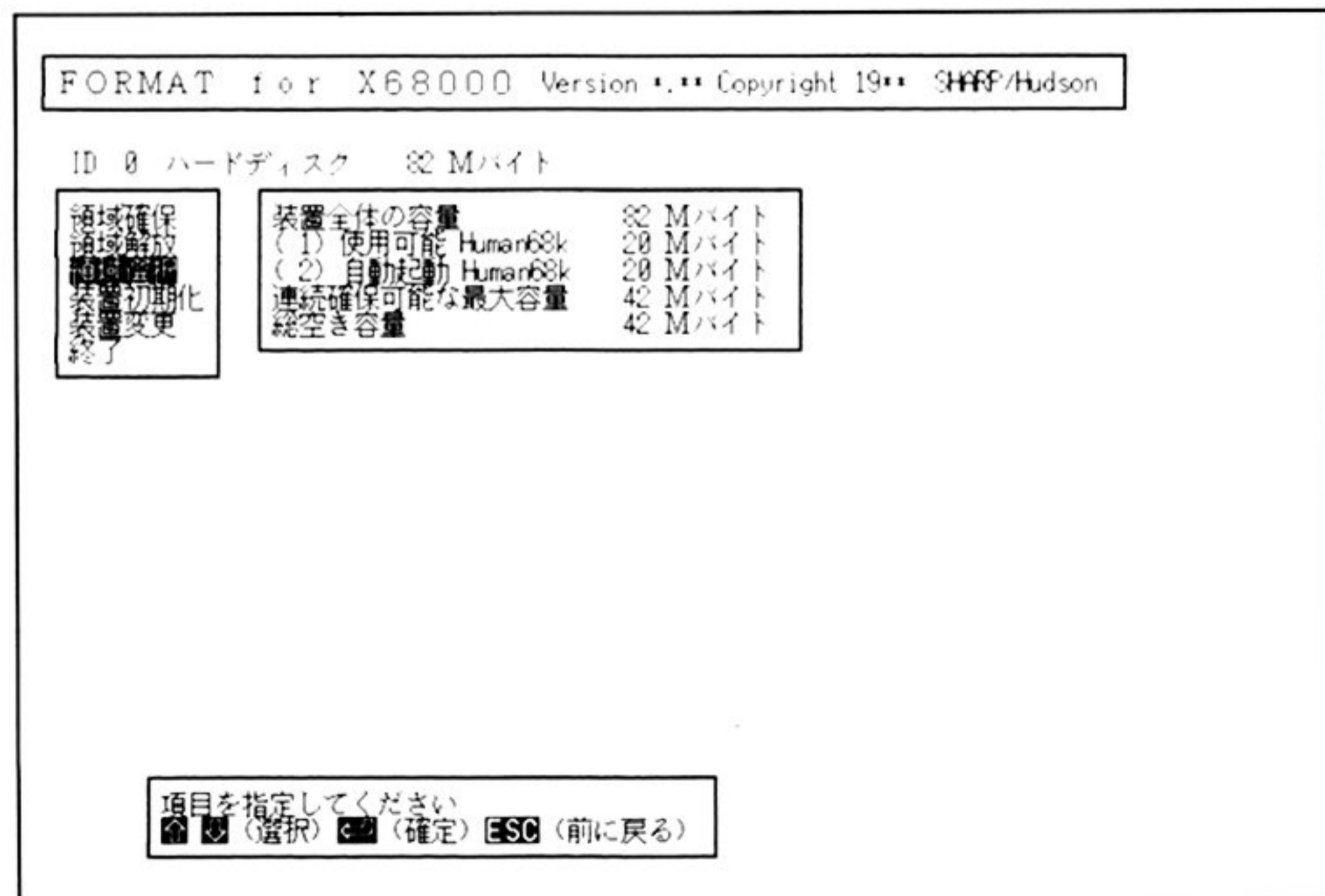
・使用可能・・・指定された領域を使えるようにします。
システムで使用可能な領域です。各使用可能領域は、システムで別々のドライブとして取り扱うことができます。

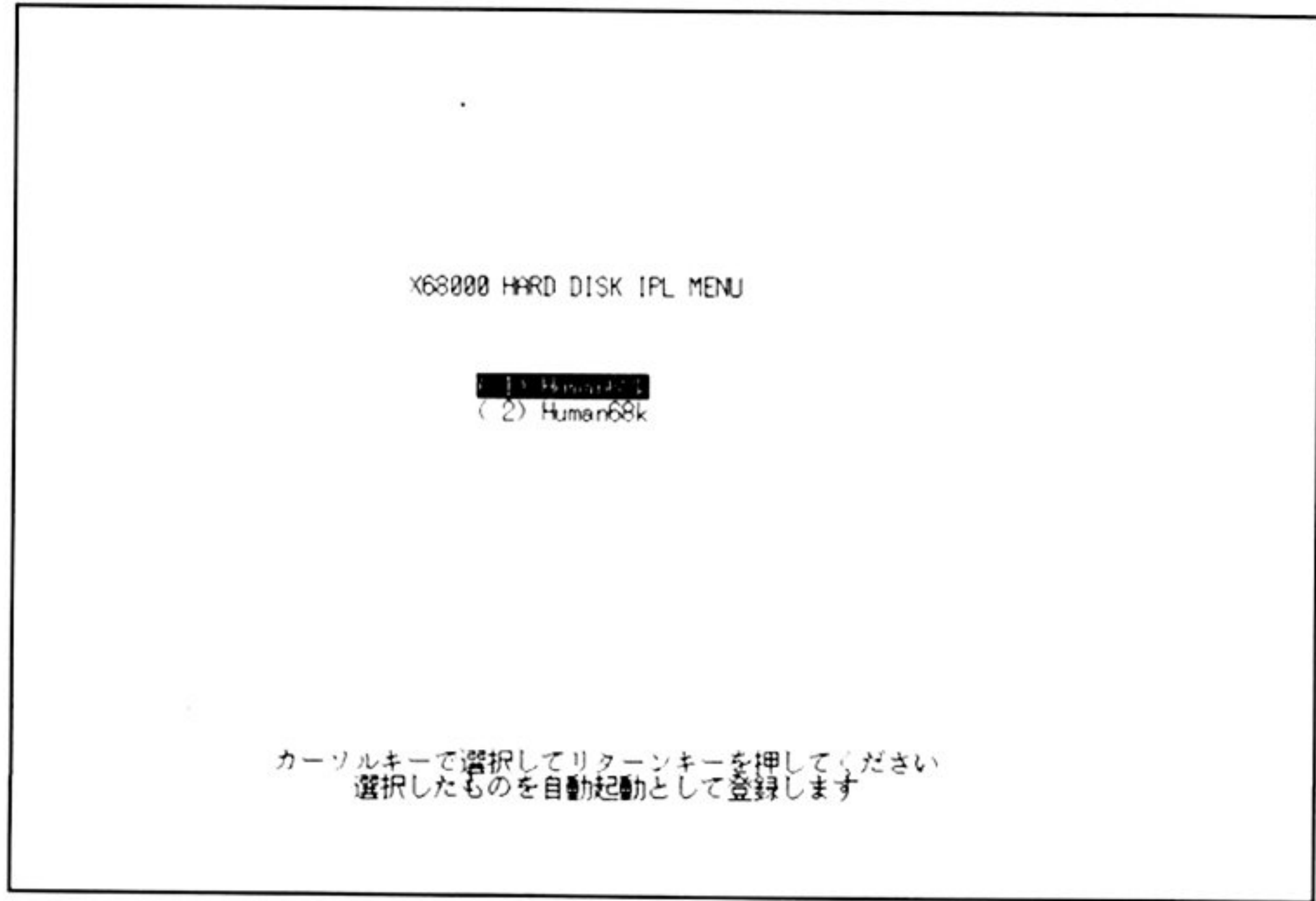
・使用不可・・・指定された領域を使えないようにします。
システムで使用しない領域です。

「領域選択」にカーソルを合わせ、↵を押すとカーソルがブロックに移動します。

ここで、領域選択する領域を選び、[⏎]を押すと、上記の3つの状態を選ぶメニューが表示されます。「自動起動」「使用可能」「使用不可」から選んで[⏎]を押してください。

この場合は、ブロック(1)から自動起動されるようになっています。ブロック(2)から自動起動させるには、次のようにします。

「領域選択」にカーソルを合わせ[⏎]を押すと、ブロックにカーソルが移動します。ブロック(2)のところで[⏎]を押すと、上と同じようなメニューが表示されます。

カーソルを「自動起動」に合わせ[⏎]を押すと、次のようにブロック(2)が「自動起動」になり、ブロック(1)が「使用可能」に変わります。自動起動できるブロックはひとつだけなので、ブロック(1)が自動的に「自動起動」から「使用可能」に変わります。

ハードディスクから起動すると、自動起動領域から起動されます。

なお、複数のブロックを自動起動および使用可能にしてあるとき、あらかじめ自動起動に設定されているブロックからではなく、別のブロックから起動させるには、HELPを押し続けながらリセットスイッチを押してください。次のような「X68000 HARD DISK IPL MENU」が表示されます。

ここで、起動したい領域をブロック番号から判断した上、その領域（ブロック）にカーソルを合わせ ⏎ を押すと、選んだ領域から自動起動されます。また次に起動するときは、今回選択した領域から自動起動されます。

なお、ドライブの割り当て状態を確認したいときは、FORMATコマンドを終了し、

　　DRIVE　／V ⏎

を実行してください。

フロッピーディスクからの起動

ハードディスクからの起動に設定(SWITCHコマンドのBOOTがSCSI 0 などになっている場合）されている場合に、フロッピーディスクドライブから起動したいときは、OPT.1 を押したままリセットスイッチを押し、OPT.1 をしばらく押し続けてください。

また、ハードディスクに何らかの異常が発生して、ハードディスクから起動できなくなったときにもOPT.1を使ってフロッピーディスクから起動してください。

OPT.1を使って起動できるプログラムは、X68030シリーズ用のプログラムです。

●装置初期化

ハードディスク全体を初期化します。ここで、

装置全体を初期化します　よろしいですか？
[Y]（実行）[N]（前に戻る）[ESC]（前に戻る）

と表示されますので、[Y]を押すと初期化が開始されます。初期化が終了すると、元のメニュー画面に戻ります。

●装置変更

フォーマットを行うID番号を変更します。SCSI装置のメニューが画面に表示されますので、フォーマットするSCSI装置をカーソルキーの[↑][↓]で選択して[↵]を押してください。

●終了

FORMATコマンドを終了します。

2．コマンド行からの実行

コマンド行で[書式]に従ったパラメータを設定した場合、メニュー画面に入らずにコマンドを実行します。コマンド行からは、フロッピーディスクのフォーマットのみ可能です。

FORMAT コマンドで指定できるスイッチは、次の3つです。

●／S

システムのコピーを行います。フロッピーディスクのフォーマット後、カレントドライブから Human68k システムを読み込み、フォーマットされたフロッピーディスクに書き込みます。

●／C

すでに物理的にフォーマットされているフロッピーディスクの管理領域、すなわちFAT 部およびディレクトリ部のみを初期化します。データ領域を物理フォーマットしないで、速くフォーマットを行うことができます。ただし、FORMAT コマンドで一度は物理フォーマットされたフロッピーディスクに対してしか使用できません。

●／V

ボリュームラベルを指定する場合に使用します。このスイッチを指定しない場合、Human68k というボリュームラベルがつけられます。

FORMAT コマンドは、たとえば次のように実行します。

FORMAT　B：[↵]

まず、次のようなメッセージが表示されます。

ドライブ B：(2HD ディスク)を初期化します。何かキーを押してください

ここで、フロッピーディスクをドライブBにセットし、何かキーを押すとフォーマットが始まり、次のようなメッセージが表示されます。

初期化中です…

フォーマットが終了すると、次のようなメッセージが表示されます。

初期化を終了しました

なお、／V スイッチをつけていれば、ここで

ボリューム名を半角21文字以内で指定してください：

とたずねてきますので、指定どおり半角21文字以内の名前を入力し、[↵](リターンキー)を押してください。

また、／S スイッチをつけていれば、次のメッセージも表示されます。ここで、d:には、カレントドライブ名が表示されます。

d：のシステムファイルを B：に転送します

これでフォーマット完了です。次のメッセージが表示されます。

他のディスクを初期化しますか？ <Y／N>

ここで[Y]を押すとフロッピーディスクがイジェクトされ、最初のメッセージに戻ります。[N]を押すとフォーマットコマンドが終了し、プロンプトに戻ります。

HIS

内部コマンド

[機　能]　ヒストリ行列の表示

[書　式]　HIS　〔<開始行>〕〔, <終了行>〕〔／B〕

[解　説]　HIS コマンドは、ヒストリ行列を表示するときに使います。

HIS [↵]

と入力すると、たとえば次のようにそれまでに実行したコマンドが順に表示されます。

```
00005 : DIR    B :
00004 : TYPE    B : FILE. 100
00003 : COPY    B : FILE. 100    A : FILE. OK
00002 : DIR
00001 : DEL    B : FILE. 100
00000 : HIS
```

左側にはいま入力した〝HIS〟コマンドから逆順に数えて〝いくつ前の〟コマンド行であるかが数字で示され、〝：〟以下右側に、保存されているコマンド行が表示されます。

〈開始行〉〈終了行〉には、表示したいヒストリ行列の始めの行、および終わりの行を指定します。どちらが大きくてもかまいません。たとえば、

HIS　20, 10 ↵　　または、HIS　10, 20 ↵

とすると、ヒストリ行列の20行から10行までを行番号付で表示します。

〈開始行〉を省略すると、最も古いヒストリ行が、〈終了行〉を省略すると、最も新しいヒストリ行が、それぞれ指定されたものとみなされます。たとえば、

HIS　,10 ↵

では、最も古いリスト行から10行までが、

HIS　20, ↵

では、20行から最も新しい行までが、それぞれ行番号付きで表示されます。

〈開始行〉のみを指定すると、そのヒストリ行だけが行番号付きで表示されます。

HIS　15 ↵

では、15行だけが番号付きで表示されます。

HIS コマンドに／B を付けて実行すると、ヒストリ行列を、行番号なしで表示します。また、ヒストリ行列にコントロール文字が含まれていても、そのまま出力します。

したがって、／B 付きの HIS コマンドの出力をリダイレクトを用いてファイルへ出力すれば、そのままバッチ処理を作成することができます。たとえば、

HIS　20,10　／B　> TEST. BAT ↵

とすれば、過去に実行した、ヒストリ行列の20行から10行までのコマンド行が、そのまま TEST. BAT ファイルに収納され、バッチファイルとすることができます。

なお、〈開始行〉および〈終了行〉で指定するヒストリ行列の行番号は、実行しようとする HIS コマンドを最も新しい行(第０行)として追加してから新しく付け直された行番号として扱われます。したがって、一度 HIS コマンドを実行し、そこで表示されたヒストリ行列の行番号をもとに次の HIS コマンドを実行すると、１つだけ行番号がずれることになります。注意してください。

注意：カンマ(,)の前後にスペースやタブを入れないようにしてください。

KEY

外部コマンド

機　能　　ファンクションキーなどの設定

書　式　　KEY

解　説　　KEYコマンドは、ファンクションキーやカーソル移動に使用するキーなどの割り当て状況が記録されているファイル(キー設定ファイル)の内容を更新したり、ファイルの内容をシステムに登録する場合に使用します。

KEY ⏎

と入力すると、次のメッセージが表示されます。

更新ですか、登録ですか？〔U／L〕

1．キー設定ファイルの更新

キー設定ファイルの内容を変更するときは、Uを押します。続いて、

入力ファイル名〔KEY. SYS〕:

が表示されます。ここで入力ファイル名を入力します。続いて、

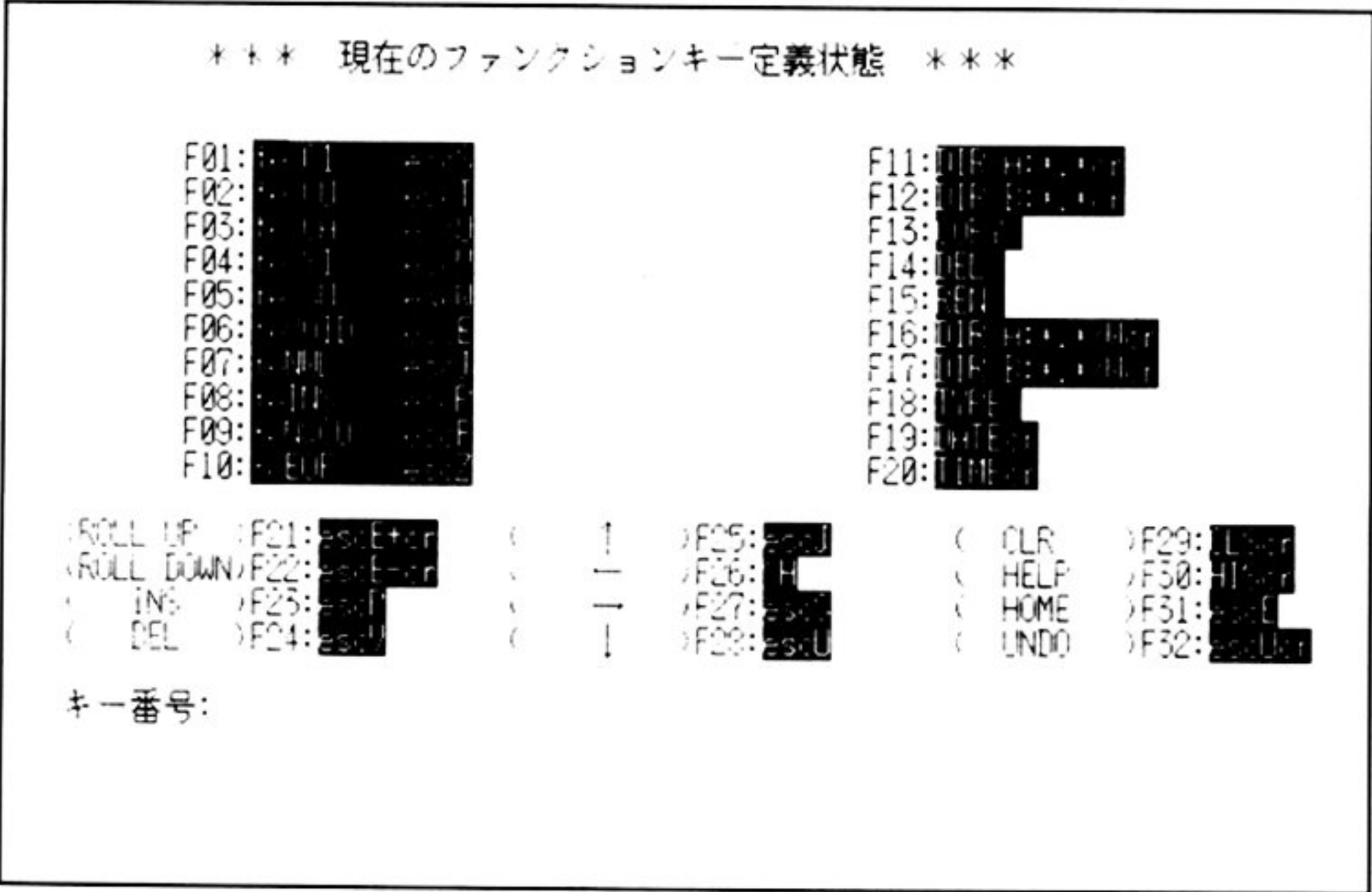
出力ファイル名〔KEY. SYS〕:

が表示されます。ここで書き出したいキー設定ファイル名を入力します。いずれもリターンキーのみを押すと自動的にKEY. SYSが対象ファイルとなります。

また、ヒストリデバイスドライバ(HISTORY.X)用のキー割り当てファイルKEY.HISもKEYコマンドで作成できます。詳しくは第6章を参照してください。

出力ファイル名を入力すると、

処理を開始しますか？〈Y／N〉

とたずねてきます。ここで、Nを押すと、処理は終了し、コマンドモードにもどります。Yを押すとKEY. SYSの中で割り当てられているキーの一覧表が次のように表示されます。

次に、

キー番号：

とたずねてきますので割り当てるキーを入力します。割り当てるキーを入力する範囲は、３文字の記号で表されます。記号は次のとおりです。

キ　ー	キー番号
F・1 ∫ F・10	F 01 ∫ F 10
SHIFT＋F・1 ∫ SHIFT＋F・10	F 11 ∫ F 20
ROLL UP	F 21
ROLL DOWN	F 22
INS	F 23
DEL	F 24
↑	F 25
←	F 26
→	F 27
↓	F 28
CLR	F 29
HELP	F 30
HOME	F 31
UNDO	F 32

キー番号を入力すると、キー機能をたずねてきます。割り当てる内容を以下の条件で入力してください。

F01～F20 半角31文字以内
その他のキー 半角5文字以内

ここでFORMAT、DISKCOPYなどのように割り当てる内容を入力しリターンキーを押してキーの決定をします。ここでそのままリターンキーを押すとすでに割り当てられていた機能がクリアされます。再びリターンキーを押すと、

終了しますか？〔Y／N〕

と表示されます。ここで、[N]を押すと、キー番号を入力する画面にもどります。[Y]を押すと、

<ファイル名>を更新しますか？〔Y／N〕

とたずねてきます。[Y]を押すとファイルを更新した後、処理を終了します。[N]を押すと、更新前の状態で、終了します。

なお、CRコードなどの特殊コードをキーに割り当てたいときは、次のようなキーを使用してください。

CRコード(リターン) →[CTRL]+[ろ] (かなキー)
BSコード(バックスペース) →[CTRL]+[む] (かなキー)
＄FE →[CTRL]+[へ] (かなキー)

ここで、＄FEは、一種の編集コードで、これに続く7文字は、画面最下行のファンクションキー表示に表示されます。8文字目以降がファンクションキーのコードとして扱われます。

また、ヌルコード(＄00)と＄1C～＄1Fのコードをファンクションキーなどに割り当てることはできません。

2. キーの登録

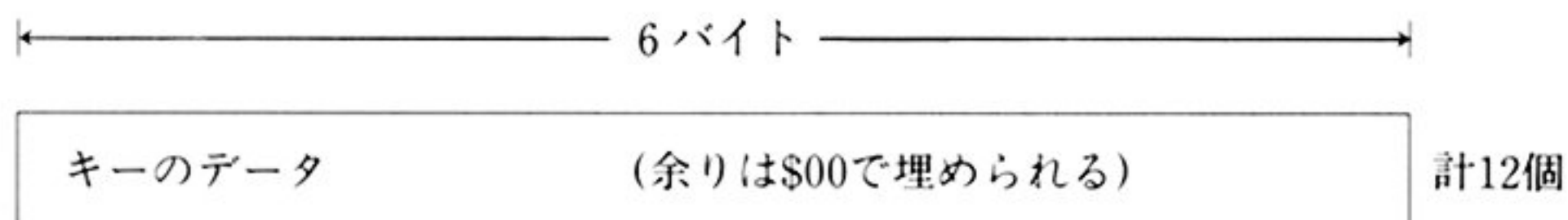
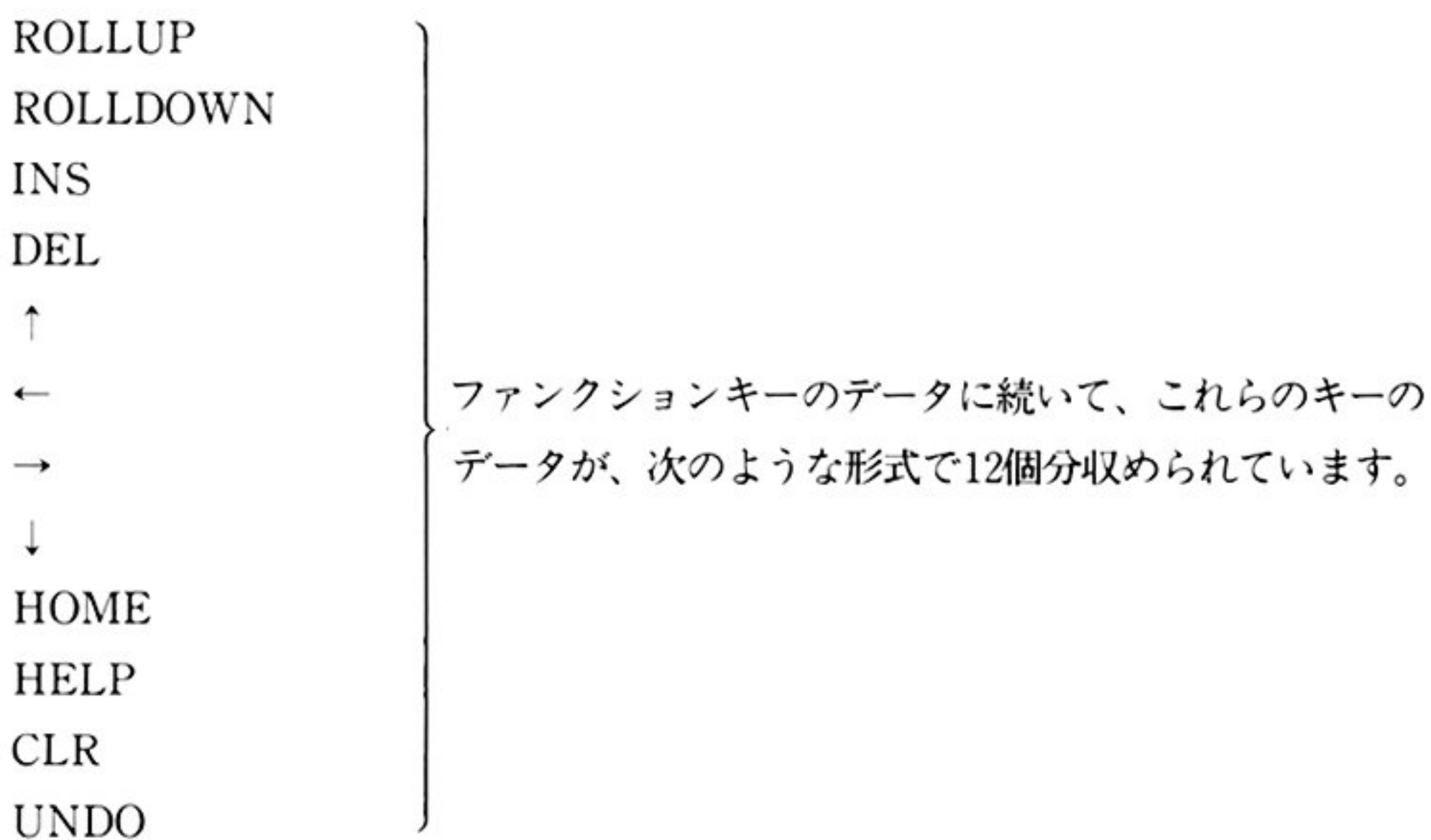
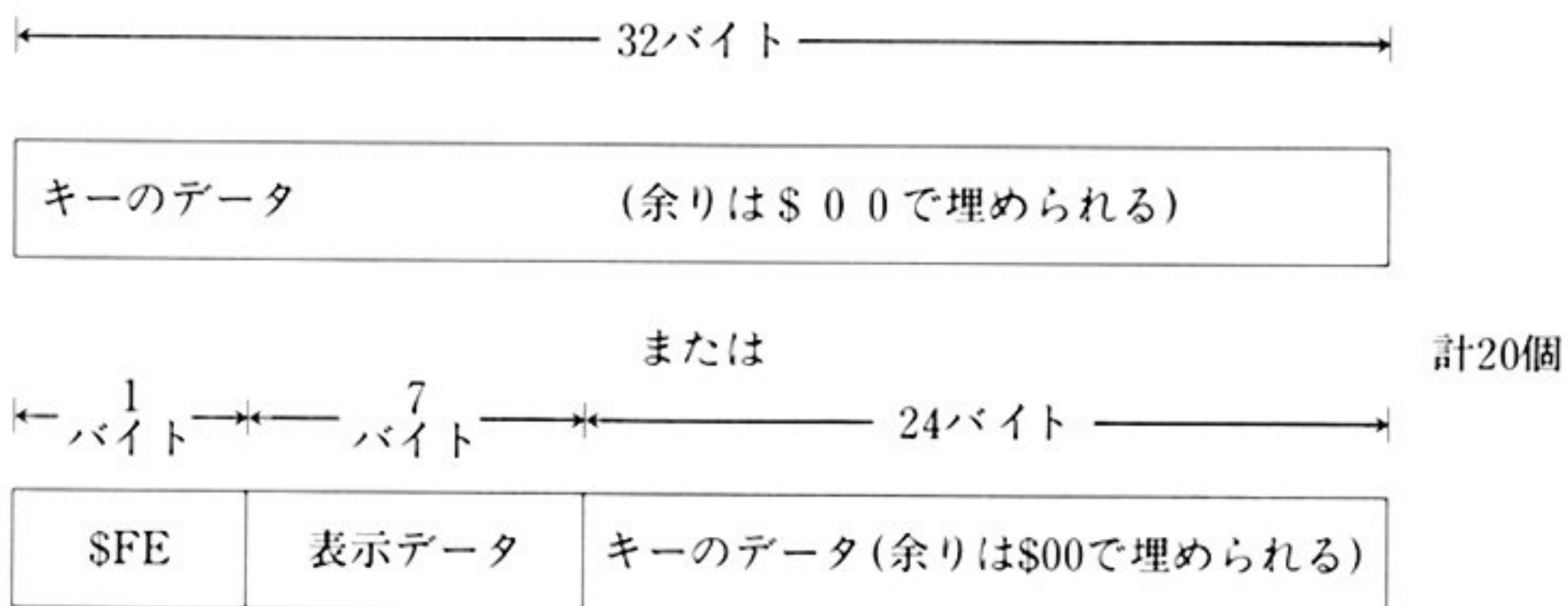
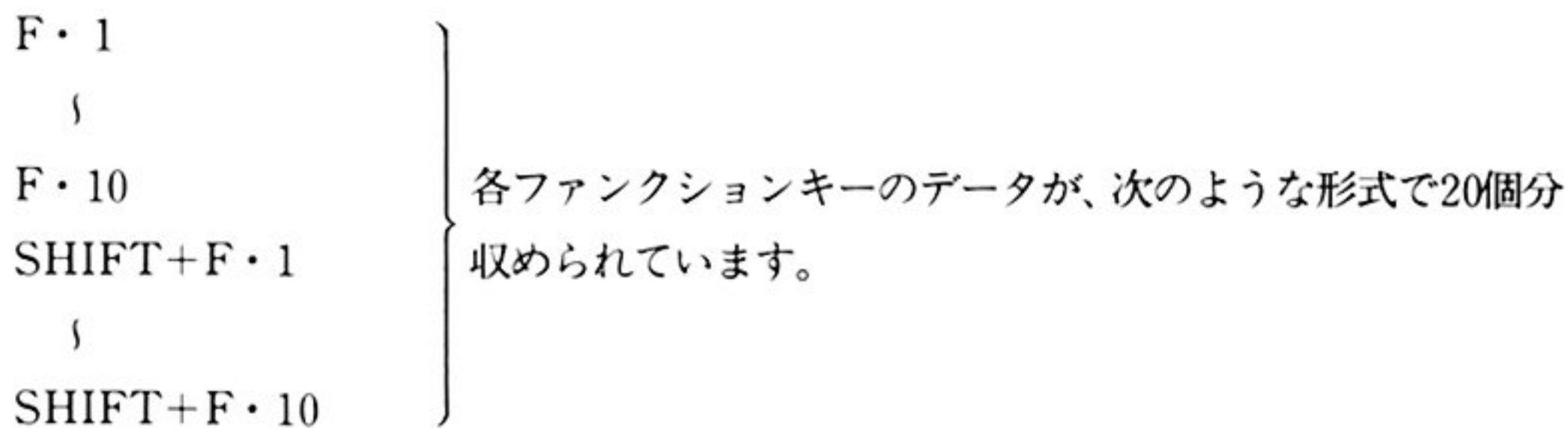
キー設定ファイルの内容をシステムに登録するときは、[L]を押します。続いて、

入力ファイル名〔KEY.SYS〕:

と表示されます。ここで、登録したいキー設定ファイル名を入力します。そのままリターンキーを押すと自動的に、KEY.SYSがシステムに登録されます。これで次のメッセージが表示されキーの登録が終了します。

ファイルの内容をシステムに登録しました

3．キー設定ファイルの内容

キー設定ファイルの内容は、次のようになっています。

MEMFREE

内部コマンド

機　能　使用可能なメモリサイズの表示

書　式　MEMFREE

解　説　現在使用できるメモリサイズを表示します。このメモリサイズはメインメモリのうち、RAMディスク、デバイスドライバなどの常駐プログラム、Human68k、親プロセスなどが利用しているメモリを除いたメモリサイズのことです。

MEMFREE ↵

と入力すると、

```
15554バイト：最大使用可能メモリ
```

というようにメモリサイズを表示します。
また、

```
14296バイト：最大使用可能メモリ
15548バイト：全体使用可能メモリ
```

というように表示された場合は、一度に確保できるメモリが14296バイトであることを示しています。

MKDIR(MD)

内部コマンド

機能	新しいディレクトリの作成
書式	MKDIR 〔<d：>〕<パス名>
別表記	MD

解説

MKDIRは、新しいディレクトリを作成するのに使うコマンドです。

<パス名>には、新しく作成するディレクトリの名前を指定します。ルートディレクトリから始まる「フルパス名」で指定(絶対指定)しても、新しいディレクトリの名前だけを指定(相対指定)してもかまいません。

たとえば、

```
MKDIR ¥SUB 1 ⏎
```

とすると、ルートディレクトリに¥SUB 1という名前のサブディレクトリが新規に作成されます。同様に、

```
MKDIR ¥SUB 1 ¥SEC 1 ⏎
```

とすると、ディレクトリ¥SUB 1の下に¥SEC 1という名前のサブディレクトリが作成されます。

また単に、

```
MKDIR DOCS ⏎
```

とすると、カレントディレクトリの下に、DOCSという名前のサブディレクトリが作成されます。

<パス名>の前にドライブ名を指定することもできます。この場合、指定したドライブ上に新しいディレクトリが作られます。

なお、作成しようとするディレクトリの名前と同名のファイルやディレクトリがすでにディスク上にある場合、ディレクトリの作成を行うことはできません。

MORE

外部コマンド

機　能　　1画面ごとの表示

書　式　　MORE　〔〈ファイル名〉〕

解　説

MOREコマンドはフィルタコマンドです。標準入力からの入力を1画面ずつに区切り、標準出力(通常は画面)に出力します。

1画面分を表示してなおデータが残っている場合は、画面の最下行に、

```
—more—
```

と表示されます。このとき、何か任意のキーを押すと、次の1画面を表示します。こうしてMOREへの標準入力データがなくなるまで表示を続けます。

MOREコマンドはフィルタコマンドですから、通常のコマンドのようにコマンド名の後にパラメータとしてファイル名を指定しても無視されます。必ずリダイレクト、またはパイプとともに使用してください。

たとえば、

```
TYPE　MSTFL. 100 | MORE ⏎
```

と入力すると、ファイル〝MSTFL. 100〟の内容を、画面に1画面分ずつ表示します。これは、次のようにしても同じ結果が得られます。

```
MORE < MSTFL. 100 ⏎
```

なおMOREコマンドは〈ファイル名〉を指定することも可能です。たとえば、

```
MORE　MSTFL. 100 ⏎
```

と入力すれば、上記2例と同様の結果が得られます。

MOVE

外部コマンド

機　能　ファイルの移動

書　式　MOVE　{〈d：〉|〈パス名1〉|〈ファイル名1〉}〔〈パス名2〉〕

解　説

〈パス名１〉または〈ファイル名１〉で指定したファイルを、他の〈パス２〉で指定したディレクトリへ移動します。移動ですから、移動元のファイルはなくなります。

〈ファイル名１〉にはワイルドカードが使用できます。

ただし、読み出し専用属性、システム属性のついているファイルは移動されません。

また、移動先のディレクトリは前もって作成されていなければなりません。

●ファイル単位の移動

```
MOVE　TEST.TXT　¥TMP ↵
```

カレントドライブのカレントディレクトリにある TEST.TXT を、カレントドライブのディレクトリ¥TMP へ移動します。

```
MOVE　*.TXT　¥TMP ↵
```

カレントドライブのカレントディレクトリにあるすべての〝*.TXT〟を、カレントドライブのディレクトリ¥TMP へ移動します。

```
MOVE　¥TMP¥TEST.TXT ↵
```

カレントドライブのディレクトリ¥TMP にある TEST.TXT を、カレントドライブのカレントディレクトリへ移動します。

●ディレクトリ単位の移動

```
MOVE　¥TMP　¥TEST ↵
```

カレントドライブのディレクトリ¥TMP のすべてのファイルを、カレントドライブのディレクトリ¥TEST に移動します。

```
MOVE　¥TMP ↵
```

カレントドライブのディレクトリ¥TMP のすべてのファイルを、カレントドライブのカレントディレクトリに移動します。

PATH

内部コマンド

機 能　外部コマンド検索のための PATH の設定

書 式　PATH 〔；〕〔〔<d：>〕<パス名>〔；〔<d：>〕<パス名>〕…〕

解 説

PATH コマンドは、外部コマンドを実行するときに、どのドライブのどのディレクトリからそのコマンドファイルをさがしてくるか(これを PATH と呼びます)を指定するものです。

PATH が設定されていないと、COMMAND. X は外部コマンドをカレントディレクトリからしかさがしません。

PATH コマンドによって PATH が設定されると、何かの外部コマンドがコマンド行に入力されたとき、COMMAND. X は、その外部コマンドのファイルをまずカレントディレクトリでさがし、そこにないと設定された PATH に従って、順にそれぞれのディレクトリ中をさがします。したがって、カレントドライブ、カレントディレクトリがどこであっても、外部コマンドの主ファイル名さえ入力すればそのまま実行することができるようになります。

PATH コマンドは、複数のディレクトリを指定することができます。この場合、次のように各ディレクトリのパス名をセミコロン(；)で区切って指定します。

```
PATH　A：¥；A：¥BIN；B：¥ ↵
```

PATH として設定するディレクトリのパス名は、ドライブ名付きのフルパス名(絶対指定)にしておくとよいでしょう。

PATH コマンドを実行すると、以前の PATH は消去され、新しく設定した PATH が有効となります。

なお、PATH コマンドをパラメータなしで入力したときは、現在の PATH が表示されます。

また、PATH の設定を解除する場合は、

```
PATH　； ↵
```

と入力してください。

例

PATH　A：¥ ↵

……カレントディレクトリの他に、ドライブＡのルートディレクトリからも外部コマンドをさがすようになります。

PATH　A：¥NAME；B：¥ ↵

……カレントディレクトリの他に、ドライブＡのサブディレクトリ¥NAMEおよびドライブＢのルートディレクトリからも外部コマンドをさがすようになります。

PATH ↵

……現在のPATH設定を表示します。

PR

外部コマンド

機　能　ファイル内容のページング

書　式　PR　〔／W〈n〉〕〔／L〈n〉〕〔／H〈ヘッダ文字〉〕〔／B〈タブサイズ〉〕〔／F〕〔／T〕〔／N〕〔〈ファイル名〉〔〈出力ファイル名〉〕〕

解　説

PR コマンドは、指定したファイルの内容をページ単位に整形するために使用します。プリンタ出力用として利用することにより、印刷を美しく行うことができます。

〈ファイル名〉には、ページ整形したいファイルのファイル名を指定します。ワイルドカードを使用することもできます。

〈出力ファイル名〉には、ページ整形した結果を収めるファイル名を指定します。PRN を指定すれば、直接プリンタに印刷することができます。〈出力ファイル名〉を指定しない場合は、結果は標準出力（通常は画面）に出力されます。

なお、PR コマンドはフィルタとして用いることもできます。

PR コマンドで使用できるスイッチは、次のとおりです。

●／W〈n〉

1行の文字数を指定します。〈n〉は指定する文字数を表します。デフォルト値は80です。

●／L〈n〉

1ページの行数を指定します。〈n〉は指定する行数を表します。デフォルト値は66です。

スイッチ／W とスイッチ／L を両方指定することにより、1ページのフォーマットを指定することができます。

●／H〈ヘッダ文字〉

ヘッダ文字を指定します。これを指定すると、各ページの上部のヘッダが付加されます。ヘッダ文字を指定しないときは〈ファイル名〉がヘッダとして付加されます。

●／B〈タブサイズ〉

水平タブのサイズを指定します。〈タブサイズ〉には2、4、8、16のうちのどれかを指定します。デフォルト値は8です。

●／F

ページの終りにフォームフィードコード（FF）をつけます。これを指定すると、／W および／L で指定した1ページのフォーマットの最後に、改ページ用としてフォームフィードコードをつけます。デフォルト値は、キャリッジリターン／ラインフィード

(CR／LF)コードを１ページの行数に合うように複数個つけます。

●／T

ヘッダ文字をつけないで出力します。

●／N

行番号表示を指定します。これを指定すると、対象ファイルの各行に順に行番号がつけられます。

PROCESS

外部コマンド

機　能

プロセスの情報を表示

書　式

PROCESS 〔／A｜／B〕

解　説

現在のプロセスがどのようにメモリを割り当てているかを表示します。

また、並行処理を行っているときのバックグラウンドプロセスや全プロセスの情報を表示します。PROCESSコマンドでは、以下のスイッチが使用できます。

／A

バックグラウンドプロセスを含め、すべてのプロセスの情報を表示します。

／B

バックグラウンドプロセスの情報だけを表示します。なお、バックグラウンドプロセスは、CONFIG.SYSに〝PROCESS〟の指定がないと実行できません。詳しくは「第7章　システムの構築」を参照してください。

●バックグラウンドプロセスの情報を表示する

PROCESS ／B ⏎

これを実行すると、たとえば次のように表示されます。

```
-----------------------------------------------------------------
PSP ID ID  プログラム名      ﾓｰﾄﾞ ｽﾘｰﾌﾟ  SSP    USP    ﾊﾞｯﾌｧ  長さ
------ -- ----------------- -- ------ ------ ------ ------ ------
0550D0 00 Human68k system   02 000000 059E84 059E84 00DA52 000080
0414B0 01 SYSTEM TIMER V1   -- 0000E7 043252 044A4E 0419BA 000080
```

それぞれの意味は次のとおりです。

- **PSP ID**

バックグラウンドプロセスのプロセス切り替え識別番号。

- **ID**

バックグラウンドプロセスの管理番号。プロセスの制御はこの番号で指示します。

- **プログラム名**

バックグラウンドプロセスのプログラム名。

- **モード**

実行モード。数値はタイムスライス何回に1回実行するかを示します(16進数)。ただし、－－のときはそのプロセスがスリープ（停止）していることを示します。

・**スリープ**

スリープ残り時間（16進数）。次回起動されるまでの時間を示します。ただし、数値が000000でかつモードが－－のときは、他から起動されるまでスリープしています。これ以外の場合は、バックグラウンドで実行されていることを示します。

・**SSP**

プロセスのSSPレジスタの値（16進数）。

・**USP**

プロセスのUSPレジスタの値（16進数）。

・**バッファ**

プロセス間通信用バッファの先頭アドレス（16進数）。

・**長さ**

プロセス間通信用バッファの長さ（16進数）。

●全プロセスの情報を表示する。

```
PROCESS ／A ↵
```

これを実行すると、たとえば次のように表示されます。

```
-----------------------------------------------------------------
 開始   終了   長さ   モード              ファイル名
------ ------ ------ ------ ------------------------------------
006800 085FFF 07F800 SUPER  Human.sys
086010 0860FF 0000F0 MALLOC
086110 08630F 000200 MALLOC
09A480 09B56F 0010F0 USER   A:¥COMMAND.x
09B580 0A2843 0072C4 MALLOC
0A2860 0A327B 000A1C MALLOC
0A3290 0A348F 000200 MALLOC
0A34A0 0A369F 000200 MALLOC
0A36B0 40BFFF 368950 USER   A:¥BIN¥PROCESS.x

086320 09A1FD 013EDE KEEP   A:¥BIN¥TIMER.x
09A210 09A467 000258 KEEP   A:¥BIN¥VDSP.x
-----------------------------------------------------------------
PSP ID ID  プログラム名  ﾓｰﾄﾞ ｽﾘｰﾌﾟ  SSP    USP   ﾊﾞｯﾌｧ   長さ
------ -- --------------- -- ------ ------ ------ ------ ------
0A36B0 00 Human68k system 02 000000 0A8480 0A8480 00E33E 000010
086320 01 SYSTEM TIMER V1 -- 00037B 08832E 089B2A 086A0A 000004
```

上段にはすべてのプロセスの情報が表示され、下段にはバックグラウンドプロセスの情報が表示されます。上段に表示されたものの意味は次のとおりです。

・**開始**

メモリブロックの開始アドレス（16進数）。データ部またはPSPの先頭アドレスを示します。

・**終了**

メモリブロックの終了アドレス（16進数）。

・**長さ**

メモリブロックの長さ（16進数）。メモリ管理領域の16バイトは含まないデータ部のみの長さ。

・モード

メモリブロックの属性を示します。属性はそれぞれ以下のような意味です。

SUPER	スーパーバイザーモードで実行
USER	ユーザーモードで実行
MALLOC	直前に表示されたSUPER、USER、KEEPになっているプロセスによって確保されたメモリ
KEEP	常駐プロセス。バックグラウンドプロセスも含みます

・ファイル名

起動されたプロセスのパス名とファイル名。

PROMPT

内部コマンド

機 能 プロンプトの設定

書 式 PROMPT 〔<プロンプトテキスト>〕

解 説 PROMPT コマンドは、Human68k のプロンプトを設定するために使用します。

Human68k のデフォルト(すなわち、システム起動時)のプロンプトは、

```
X>
```

です。ここで、X は、そのときのカレントドライブを表す英文字1文字です。

PROMPT コマンドを使うことにより、このプロンプトを変更することができます。

<プロンプトテキスト>には、設定したい文字列を指定します。たとえば、

```
PROMPT  Human:
```

と入力すれば、プロンプトは、"Human:" に変わります。

PROMPT コマンドでは、通常の文字列のほかに、ドルマーク($)に続いて以下の記号を指定できます。これによって、特別なプロンプトを作ることができます。

N	カレントドライブ名(大文字表示)
n	カレントドライブ名(小文字表示)
D	現在の日付
T	現在の時刻
P	カレントドライブ名とカレントディレクトリ名
V	バージョン表示
$	"$" 文字
G	">" 文字
L	"<" 文字
B	"|" 文字
_	改行
S	ブランク
H	バックスペース
E	ASCIIコード"$1B"(エスケープコード)

<プロンプトテキスト>を指定せずに PROMPT コマンドを実行すると、デフォルトのプロンプトに変更されます。

例

PROMPT　NG ↵

……デフォルトのプロンプトと同じ文字に設定します。

PROMPT　Time=T_Date=D_

……Time=〈現在の時刻〉

Date=〈現在の日付〉

の2行のプロンプトを表示するように設定します。

PROMPT　$E[7 m$NGE[m

……プロンプトだけを反転表示し、その他の入力文字は通常表示するよう設定します。

RECOVER

機能　不良クラスタを含むファイルまたはディスクの修復

書式　RECOVER　{<d：>｜<パス名>〔<ファイル名>〕}

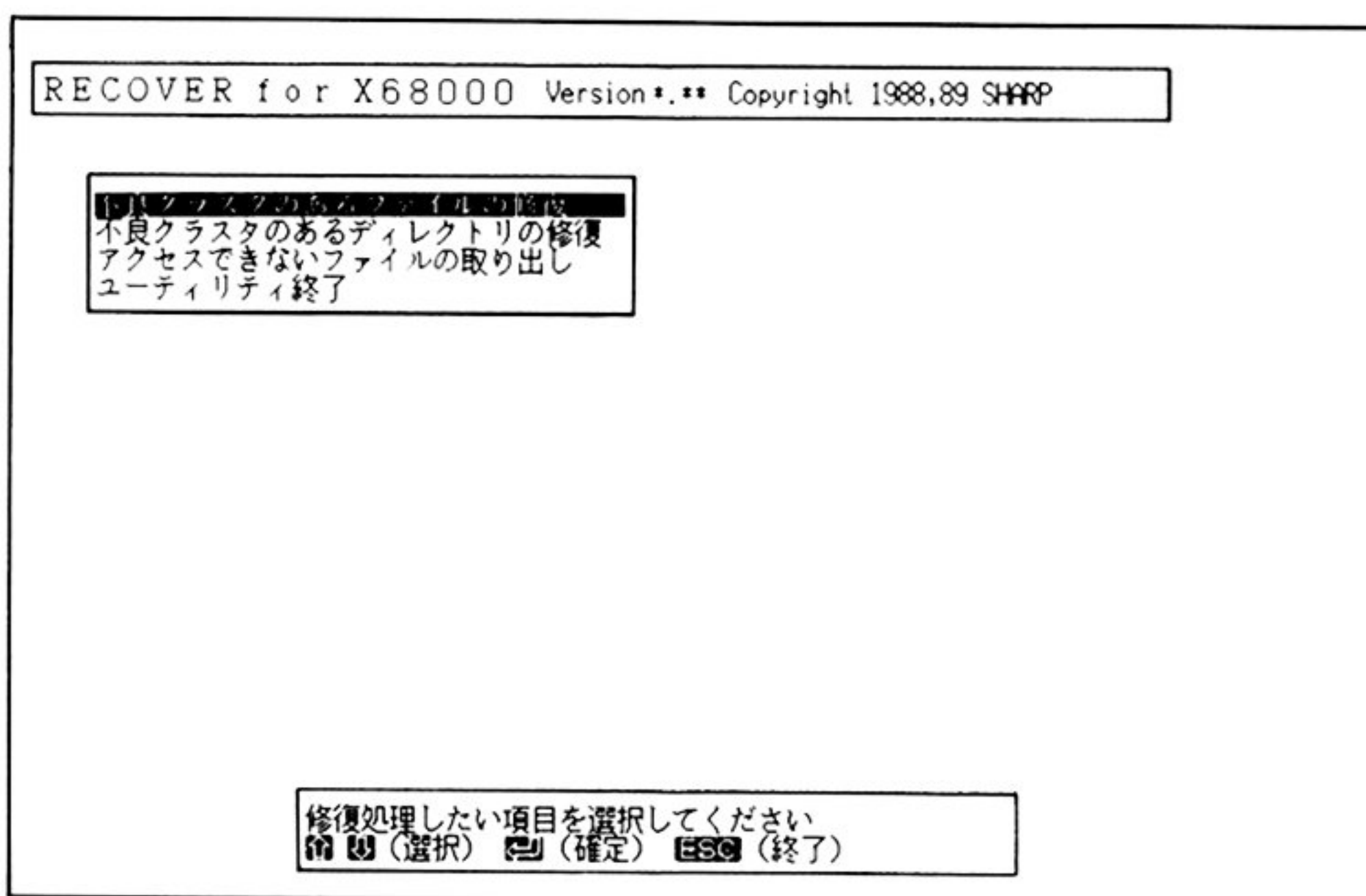
解説

ディスクに不良クラスタがある(リードエラーになった)場合には、このコマンドにより、不良クラスタのあるファイル、ディレクトリまたはディスク全体（不良クラスタが階層ディレクトリである場合など）を使用可能な状態に修復することができます。

1.メニュー画面からの実行

RECOVER ↵

と入力すると、次のようなメニュー画面が表示されます。

```
RECOVER for X68000 Version*.** Copyright 1988,89 SHARP

不良クラスタのあるファイルの修復
不良クラスタのあるディレクトリの修復
アクセスできないファイルの取り出し
ユーティリティ終了

修復処理したい項目を選択してください
↑ ↓（選択） ↵（確定） ESC（終了）
```

ここで、カーソルキーの ↑ ↓ で処理を選択して ↵ を押します。それぞれの項目は、次のようなときに使用します。

●不良クラスタのあるファイルの修復

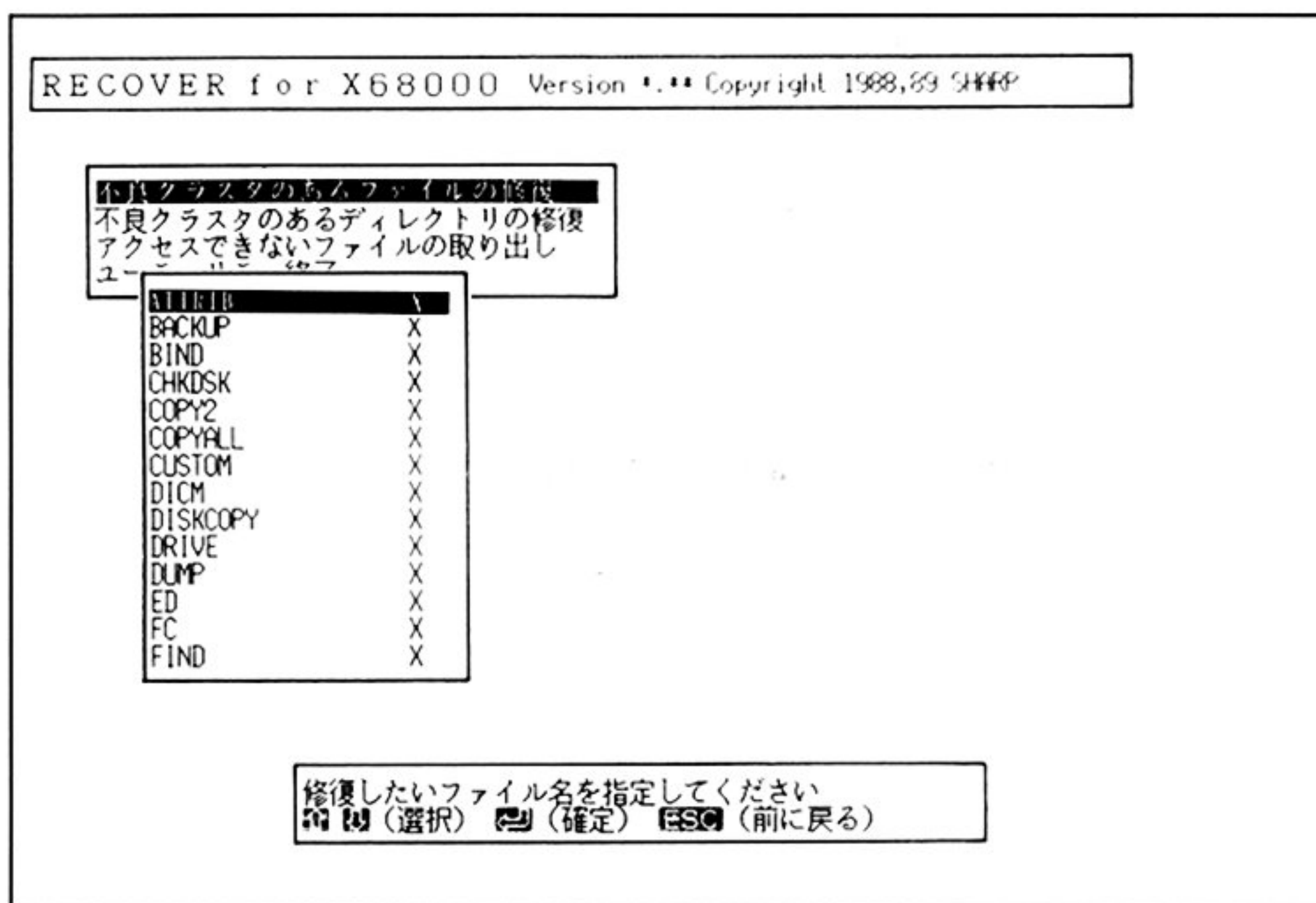
ファイルがリードエラーになったときに使用します。実行すると、ファイルの中の不良クラスタを飛ばしながらファイルを読み込み、新しいファイルを作成します。不良クラスタは記憶され、以後そのセクタを使用できないように設定します。ただし、修復されたファイルの一部は消失するので、注意してください。

●不良クラスタのあるディレクトリの修復

リードエラーになったディレクトリを修復します。ただし、カレントディレクトリの下にあるファイルの一部は消失するので、注意してください。

●アクセスできないファイルの取り出し

ディスク全体にわたって修復します。階層ディレクトリがなんらかの事故で読めなくなったときは、そのディレクトリの下にあるファイルをルートディレクトリに移動してきます。ただし、ファイル名は暫定的なものになりますので、後で変更してください。

項目の選択を終えると、ドライブ名、ディレクトリ名、ファイル名などを選択する画面になります。次はファイルを選択中の画面です。

2．コマンド行からの実行

コマンド行で 書式 に従ったパラメータを設定した場合、メニュー画面に入らずにコマンドを実行します。

ディレクトリやファイルを修復するには次のように入力します。

RECOVER　〈パス名〉〈ファイル名〉

〈パス名〉〈ファイル名〉を指定した場合にはファイル中の不良セクタを飛ばしながらセクタごとに読み込むようにします。そして不良セクタを記録し、以後そのセクタを使わないようにします。ただし、ファイルの一部分は消失しますので注意してください。〈パス名〉のみを指定した場合は、そのディレクトリを修復します。ただし、そのディレクトリ下の一部のファイルは消失しますので注意してください。

ディスク全体を修復するには次のように入力します。

RECOVER　<d：>

<d：>は修復するディスクのあるディスクドライブを指定します。階層ディレクトリがなんらかの事故で読めなくなったとき、そのディレクトリ下にあるファイルをルートディレクトリにもってきます。ただし、ファイル名は適当につけられますので、後で変更してください。

REN(RENAME)

内部コマンド

機能　ファイル名の変更

書式　REN　〈ファイル名1〉〈ファイル名2〉〔／Q〕

別表記　RENAME

解説

RENコマンドは、ファイル名を変更するために使用します。

変更する前のファイル名を先に(〈ファイル名1〉)、変更したいファイル名を後に(〈ファイル名2〉)指定します。なお、〈ファイル名1〉にはドライブ名、パス名の指定をすることができますが〈ファイル名2〉にパス名を指定する必要はありません。

● WKFL．100を WKFL．200に変更します

```
REN　WKFL．100　WKFL．200 ↵
```

●拡張子に MST を持つファイルすべての拡張子を WK に変更します

```
REN　*．MST　*．WK ↵
```

●ドライブ B にあるファイル TEMP を TEMP．100へ変更します

```
REN　B：TEMP　????．100 ↵
```

存在しないファイルのファイル名を変更しようとした場合は、

ファイルが見つからないか、ファイル名が重複しています

というエラーメッセージが表示されます。

複数のファイルのファイル名を変更する場合、スイッチ／Qをつけると変更の対象となるファイル名とともに確認のメッセージが表示されたのち、変更します。

注意

・ディレクトリ名に対してRENコマンドを使わないでください。

・ファイル名の変更を行うとき、ファイルのあるディレクトリが仮想ドライブや仮想ディレクトリを含むときは、次のことに注意してください。

仮想ドライブや仮想ディレクトリ上にあるファイルのファイル名を変更しようとすると、指定したファイル名だけでなく実体となるファイルのファイル名も同時に変更されてしまいます。たとえば、仮想ドライブの指定が

```
C：＝A：TEST
```

のとき、ドライブAとドライブCに次のようなファイルがあるときには、

コマンド行で次のようにREN コマンドを実行すると、

REN C:SAMPLE1.DAT TEST1.TXT ↵

実際にRENコマンドが行う作業は、

REN A:¥TEST¥SAMPLE1.DAT TEST1.TXT ↵

となり、実体としてのA:¥TEST上のファイル名が変更され、その結果として仮想ドライブC上のファイル名も変更されてしまいます。つまり、変更したかった仮想ドライブ上にあるファイルのファイル名だけでなく、実体のファイル名も変更されてしまいます。

逆に、実体のディレクトリ上のファイル名を変更すると、仮想ドライブや仮想ディレクトリ上のファイル名も変更されることになります。このように、ファイル名を変更するときに、仮想ドライブや仮想ディレクトリを含むディレクトリを指定すると、実体のファイルと、仮想ドライブや仮想ディレクトリ上のファイルの両方が変更されますので、ご注意ください（仮想ドライブや仮想ディレクトリについては、SUBSTコマンドをご覧ください)。

RESTORE

外部コマンド

機　能　BACKUPコマンドでバックアップされたファイルの復元

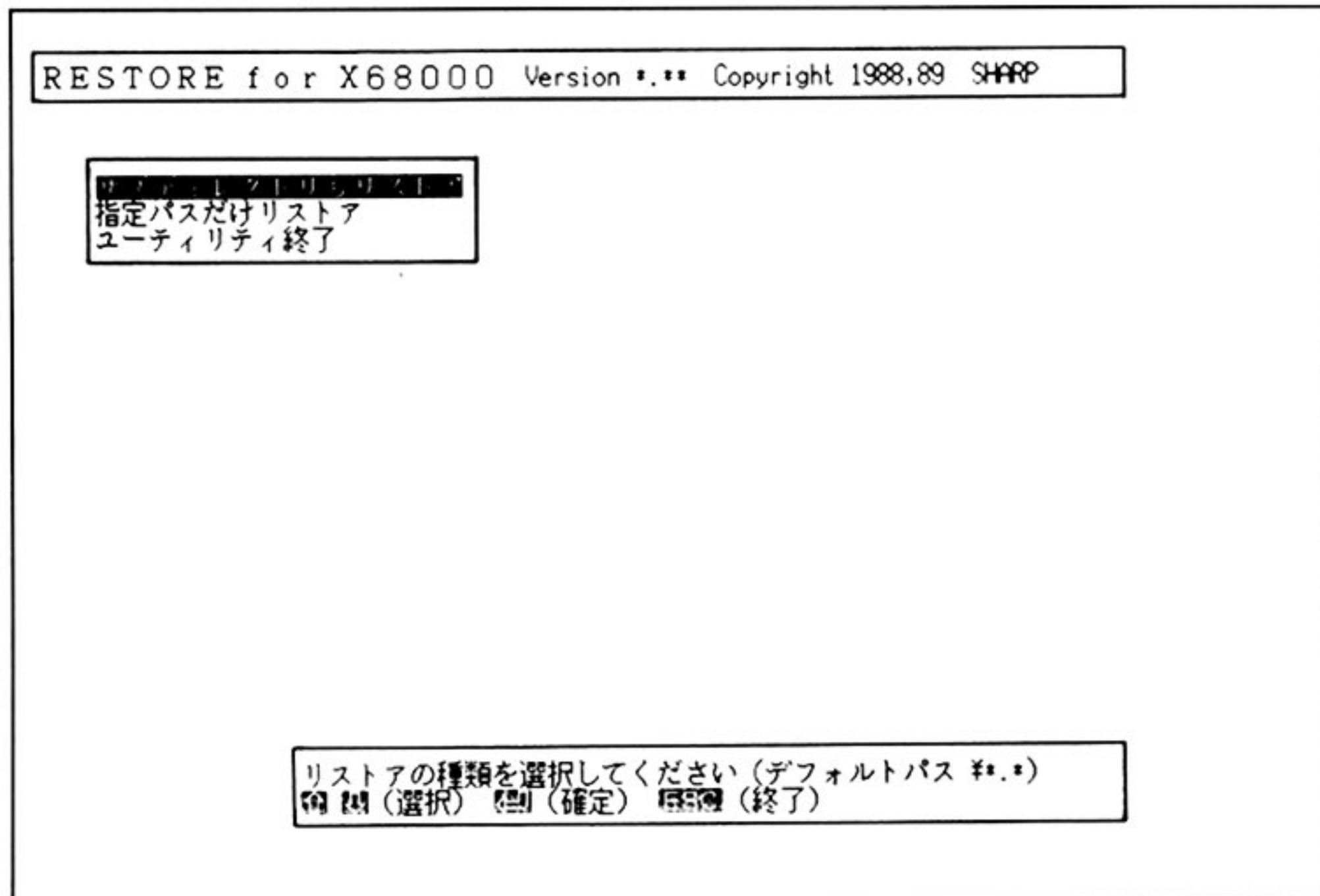
書　式　RESTORE　<fd：>　<hd：>〔<パス名>〕　〔／S〕〔／P〕〔／B：<日付>〕〔／A：<日付>〕〔／E：<時刻>〕〔／L：<時刻>〕〔／N〕

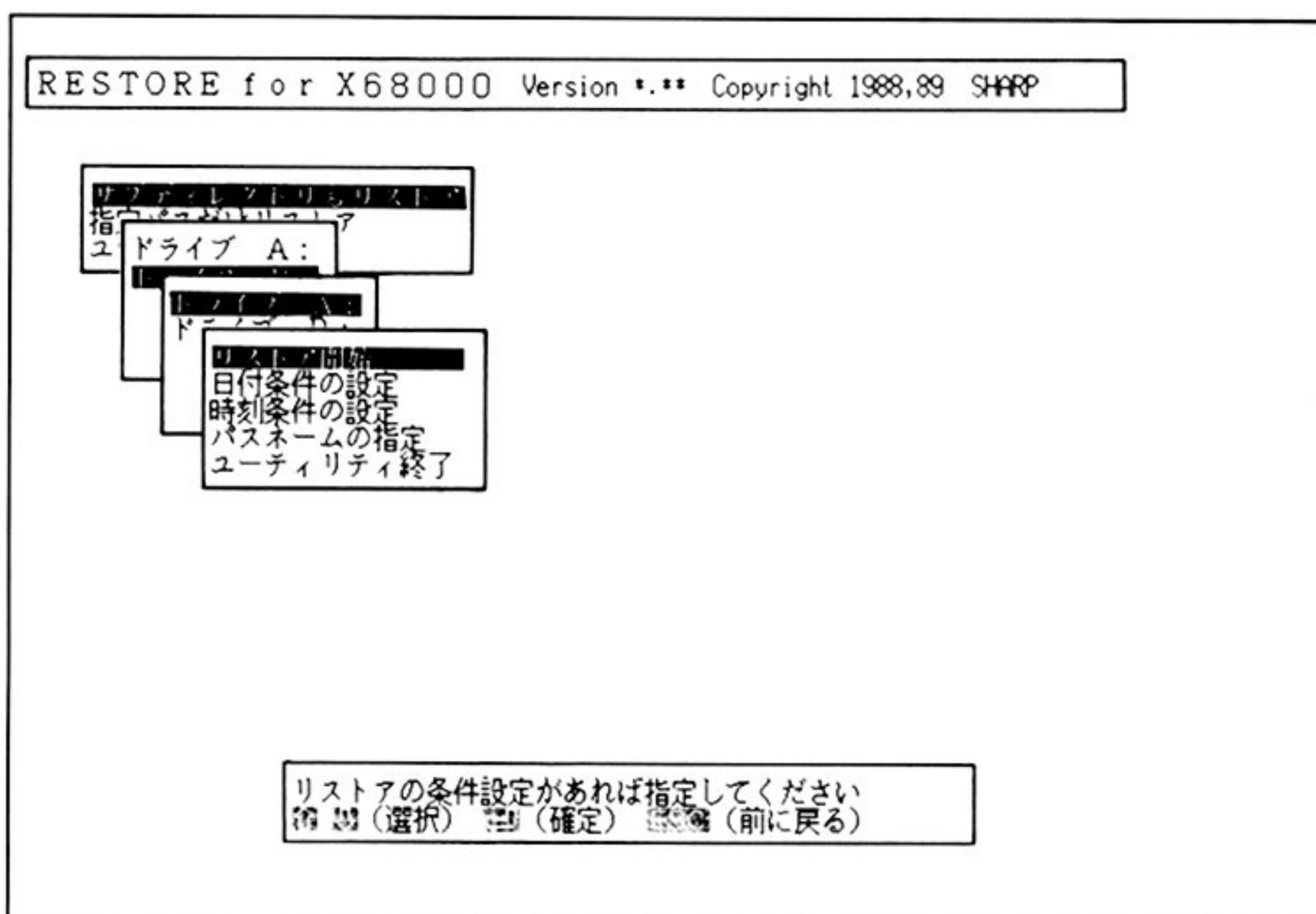
解　説　BACKUPコマンドでバックアップされたファイルを復元します。

1.メニュー画面からの実行

RESTORE ↵

と入力すると、次のようなメニュー画面が表示されます。

```
RESTORE for X68000 Version *.** Copyright 1988,89 SHARP

[illegible]
指定パスだけリストア
ユーティリティ終了

リストアの種類を選択してください（デフォルトパス ¥*.*）
↑ ↓（選択） ↵（確定） ESC（終了）
```

ここで、カーソルキーの ↑ ↓ で処理を選択して ↵ を押します。この後、画面はドライブ名やディレクトリ名を選択する画面になりますので、カーソルキーで選択してください。

たとえば、ドライブB（フロッピーディスク）からドライブA（ハードディスク）へ、ファイルを復元する場合は、次の画面のように〝サブディレクトリもリストア〟→〝ドライブB：〟→〝ドライブA：〟→〝リストア開始〟を ↵ で選択します。

必要であれば最後に日付条件（指定した日付の間に作成したファイルをリストア）や時刻条件（指定した時刻の間に作成したファイルをリストア）を設定し、「リストア開始」を選択してください。

2.コマンド行からの実行

コマンド行で 書式 に従ったパラメータを設定した場合、メニュー画面に入らずにコマンドを実行します。

最初の〈fd：〉には、バックアップされたファイルを含むフロッピーディスクの入っているフロッピーディスクドライブを指定します。次に、そのファイルを復元するハードディスクドライブを〈hd：〉で指定します。〈パス名〉が指定されると、そのディレクトリ名と一致したディレクトリ上のファイルをフロッピーディスクからさがしてハードディスクに復元します。

RESTORE コマンドでは、以下のスイッチを使用することができます。

／S　サブディレクトリも復元します。

／P　もし、ファイルの属性が、〝不可視属性のファイル〟または〝読み出し専用のファイル〟ならば、それらを復元してよいかどうかたずねてきます。

／B　指定された日付以前に変更されたファイルだけを復元します。

／A　指定された日付以降に変更されたファイルだけを復元します。

／E　指定された時刻以前に変更されたファイルだけを復元します。

／L　指定された時刻以降に変更されたファイルだけを復元します。

／N　復元先のハードディスクに存在しないファイルだけを復元します。

RMDIR(RD)

内部コマンド

機能　ディレクトリの削除

書式　RMDIR　〔〈d：〉〕〈パス名〉

別表記　RD

解説

RMDIRコマンドは、指定したディレクトリを削除するときに使用します。

〈d：〉、〈パス名〉には、それぞれ削除しようとするディレクトリが存在するドライブ名、そのディレクトリのパス名(絶対指定でも相対指定でも可能です)を指定します。

ただし削除しようとするディレクトリには、〝.〟(自分自身)と〝..〟(親ディレクトリ)以外のファイルやディレクトリがあってはいけません。対象ディレクトリのすべてのファイルを削除してから実行してください。ファイルなどが残っていると、ディレクトリは削除できません。

また、〈パス名〉には、その時点でのカレントディレクトリのフルパス名に含まれているディレクトリ名を指定することはできません。たとえば、カレントディレクトリが〝¥SUB 1 ¥SEC 1〟であるときに、

```
RMDIR　¥SUB 1 ⏎
```

や、

```
RMDIR　¥SUB 1 ¥SEC 1 ⏎
```

を実行しても、ディレクトリの削除を行うことはできません。

注意

ディレクトリを削除するとき、仮想ドライブが設定されているときには、次のことに注意してください。

削除しようとするディレクトリに仮想ドライブが設定されているときにRMDIRコマンドを実行すると、

ディレクトリが空でないか、見つからないか、カレントパスです

と表示され、指定したディレクトリを削除できません。

このように削除しようとするディレクトリに仮想ドライブが設定されているときに、そのディレクトリを削除するときは、まず仮想ドライブの設定を解除してからRMDIRコマンドを使ってください。なお、仮想ドライブにボリュームラベルが設定されているときには、仮想ドライブの設定を解除する前にボリュームラベルを削除してください(仮想ドライブや仮想ディレクトリについては、SUBSTコマンドをご覧ください)。

SCREEN　　内部コマンド

機　能　画面モードの設定

書　式　SCREEN　〔〔〈画面サイズ〉〕,〔〈グラフィックモード〉〕,〔〈表示モード〉〕〕

解　説

SCREEN コマンドは、ディスプレイ画面のモードを設定するために使用します。

〈画面サイズ〉は、画面に表示される横方向の文字数(グラフィックではドット数)を決めるもので、0または1を指定します。

0：横96文字(グラフィック768×512ドット)モード
1：横64文字(グラフィック512×512ドット)モード

〈グラフィックモード〉は、グラフィック画面で表示できる色の数を決めるもので、0～3を指定します。ただし、〈画面サイズ〉が0のときは、〈グラフィックモード〉に2および3を指定することはできません。

0：グラフィックなし
1：グラフィック16色
2：グラフィック256色(〈画面サイズ〉が0のときは無効)
3：グラフィック65536色(〈画面サイズ〉が0のときは無効)

〈表示モード〉は、画面上に何を表示するかを決めるもので、0～3を指定します。

ただし、〈グラフィックモード〉が0のときは、〈表示モード〉に1または3を指定することはできません。

0：テキストのみ表示
1：テキスト、グラフィックを表示(〈グラフィックモード〉が0のときは無効)
2：テキスト、スプライトを表示
3：テキスト、グラフィック、スプライトを表示(〈グラフィックモード〉が0のときは無効)

以上のパラメータをすべて省略して、

```
SCREEN ⏎
```

と入力した場合は、各パラメータには0が指定されたものと見なされ、

```
SCREEN  0,0,0 ⏎
```

とした場合と同じになります。

ただし、1個または2個のパラメータを省略した場合は、その省略されたパラメータに対しては、以前に指定した値が指定されたものと見なされます。たとえば、

```
SCREEN  1,2,1 ↵
```

と入力した後に、

```
SCREEN  ,,3 ↵
```

とすると、〈画面サイズ〉には1が、〈グラフィックモード〉には2が、それぞれ指定されたものとみなされ、〈表示モード〉のみが3に変更されます。

なお、グラフィック画面用のメモリをRAMディスクに使用するような設定を行った場合、〈グラフィックモード〉には必ず0（グラフィックなし）を指定するようにしてください。これ以外の値を指定すると、RAMディスクの内容がこわれてしまいます。

もし、〈グラフィックモード〉に0以外を指定してRAMディスクをこわしてしまったときには、〈グラフィックモード〉に0を指定してSCREENコマンドを実行した後、FORMATコマンドでRAMディスクをフォーマットし直してください。

注意：各パラメータの間は、必ずカンマ（,）で区切るようにしてください。
また、RAMディスクについては、「第7章　システムの構築」を参照してください。

SET

内部コマンド

機　能　　環境文字列の値の設定

書　式　　SET　〔〈名前〉=〔〈文字列〉〕〕

解　説

Human68k には「環境文字列」というものがあります。この環境文字列は、システムによって管理される領域(環境エリア)に格納されており、これを利用することで、Human68k 下で動作する複数のプログラム間で情報をやりとりすることができます。

SET コマンドは、環境エリアに、指定された〈名前〉で、指定された〈文字列〉を書き込みます。書き込まれる形式は、

　　〈名前〉=〈文字列〉

のように、〝=〟でつながれた形となります。

指定した名前がすでに環境文字列名として使われている場合は、文字列が新しい文字列に置きかえられます。

〝〈名前〉=〟だけを指定して SET コマンドを実行すると、その名前に関連した文字列は環境文字列から削除されます。

何も指定しないで SET コマンドを実行すると、現在設定されている環境文字列がすべて表示されます。PATH も環境文字列の一種ですので、このとき同時に表示されます。

環境文字列は、バッチファイルでも利用することができます。環境文字列の〝名前〟をバッチファイル中に入れておくことにより、これを置きかえ可能なパラメータとして扱うことができるのです。

たとえば、バッチファイル中に、

```
DUMP　%TEMP%
```

という行を入れておき、このバッチファイルを実行する前に、

```
SET　TEMP=FILE ↵
```

とセットすると、パラメータ%TEMP%がバッチファイルの実行時に自動的に〝FILE〟に置きかえられ、

```
DUMP　FILE
```

として実行されます。これによって、バッチファイルを再編集しなくてもパラメータ名を変更することができます。

このように、バッチファイルで名前によるパラメータを用いるときには、置きかえの対象とする文字列(名前)を%で囲むようにします。

例

SET　TTY=X68000 ⏎

……TTYの値は〝X68000〟となります。

SET　TTY= ⏎

……TTYに設定されていた文字列は削除され、TTYの項目が環境文字列からなくなります。

SET ⏎

……現在設定されている環境文字列がすべて画面に表示されます。

SORT

外部コマンド

機　能　データのソート(並べかえ)

書　式　SORT　〔/R〕〔/I〕〔/+n〕〔/T〈タブサイズ〉〕

解　説

SORT コマンドはフィルタコマンドです。標準入力からの入力データを、行単位でアルファベット順(漢字コード順)に並べかえ、標準出力に出力します。

指定できるスイッチには、次のものがあります。

●/R

逆順(Z～A)にソートします。

●/I

大文字・小文字を同じものとして扱い、ソートします。

●/+n

n 桁目から行の最後までの文字列をソートキー(ソートの対象)にします。このスイッチが指定されていないと、つねに行全体をソートの対象にします(n に 1 が指定されたのと同じです)。

●/T〈タブサイズ〉

水平タブのサイズを指定します。2、4、8、16のうち、どれかを指定することができます。デフォルト値は 8 です。/+n でソートキーの開始桁数を指定する際に、水平タブを最大いくつのブランクとして扱うのかを設定することができます。

SORT コマンドはフィルタコマンドですから、通常のコマンドのようにコマンド名の後にパラメータとしてファイル名を指定しても無視されます。必ずリダイレクト、またはパイプとともに使用してください。

次の例はファイル WKFL. AAA の各行を正順(A→Z)にソートし、結果を WKFL. BBB に書き込みます。

```
SORT < WKFL.AAA > WKFL.BBB ↵
```

次の例はディレクトリの出力の36桁目以降を対象にソートします。ディレクトリ表示の36桁目以降は各ファイルおよびディレクトリの作成・変更の日付と時刻ですから、結果的に日付および時刻の古い順にディレクトリが表示されることになります。

```
DIR | SORT /+36 ↵
```

次の例はファイル WKFL. CCC を逆順にソートし、WKFL. REV に書き込みます。

```
SORT /R < WKFL.CCC > WKFL.REV ↵
```

SPEED

外部コマンド

機 能　RS−232Cインターフェイスに対するパラメータの設定・起動

書 式　SPEED 〔<パラメータ>〕

解 説　SPEEDコマンドは、RS−232Cインターフェイスに対するパラメータの設定やその起動を行うときに使用します。標準では、本体内蔵のRS−232Cインターフェイスに対して設定を行います。

1．メニュー画面からの実行

SPEED ↵

と入力すると、次のようなメニュー画面が表示されます。

```
SPEED for X68000 Version *.** Copyright 1989 SHARP/Hudson

      ボーレート  データ長  パリティ  ストップ  フロー制御
AUX   1200bps     8         なし      1         Xon
終了

本体内蔵のRS−232Cのパラメータを設定します
(選択) (確定) (設定せずに終了)
```

この画面は、本体内蔵のRS−232Cインターフェイスに対するパラメータ設定の画面です。別売のRS−232Cボード（CZ-6BF1）を本体に装着し、しかもRS−232Cドライバ（RSDRV.SYS）が組み込まれている場合は、AUX1〜AUX3（ボード1枚）またはAUX1〜AUX5（ボード2枚）の設定値が表示されます（RS−232Cボードは2枚まで装着でき、1枚のボードに2チャンネルあります）。次の画面は、RS−232Cボードを1枚装着した場合です。このとき表示されるAUX0はカレントボードです。

SPEED for X68000 Version *.** Copyright 1989 SHARP/Hudson

	ボーレート	データ長	パリティ	ストップ	フロー制御
[AUX1]	19200bps	8	なし	1	Xon
[AUX1]	19200bps	8	なし	1	Xon
[AUX2]	19200bps	8	なし	1	Xon
[AUX3]	19200bps	8	なし	1	Xon
終了					

パラメータの設定を行うＲＳ－２３２Ｃを指定してください
↑ ↓（選択）↵（確定）ESC（設定せずに終了）

この画面では、次のキー操作が可能です。

キー	機能
ESC	SPEEDコマンドの終了。設定前の状態で終了します。
↵	各項目の入力。
↑	各項目の値をアップ
スペース	各項目の値をアップ
↓	各項目の値をダウン
↓ ↑ → ←	項目の移動

内容を変更するときは、「AUX」にカーソルを合わせ↵を押します。カーソルが各項目の欄に移動しますから、そこで↑や↓で内容を変更してください。変更した内容を変更前の内容に戻したいときは、ESCを押してください。

設定が終わったら↵を押して「AUX」に戻り、「終了」の項目にカーソルを移動して、↵を押します。画面には確認のメッセージが表示されますので、設定した内容で更新するときは Y を押してください。なお、途中で終了するときは ESC を押します。確認のメッセージが表示されますので、任意のキーを押してください。

項目については、次の説明を参照して設定してください（カッコ内は、コマンド行から実行する場合のパラメータです)。

●ボーレート

データを転送する速度 (bps) を指定します。ボーレートは、次のいずれかを指定することができます。

19200bps（19200）
9600bps（9600）
4800bps（4800）
2400bps（2400）
1200bps（1200）
600bps（600）
300bps（300）
150bps（150）
75bps（75）

●データ長

データのキャラクタの長さ（ビット数）を指定します。次のいずれかを指定することができます。

8（B 8）
7（B 7）
6（B 6）
5（B 5）

●パリティ

パリティチェックのモードを指定します。次のいずれかを指定することができます。

なし（PN）
偶数（PE）
奇数（PO）

●ストップ

ストップビットの数を指定します。次のいずれかを指定することができます。

1　(S 1)
1.5(S1.5)
2　(S 2)

●フロー制御

XON、XOFFコードまたは、RTS、CTSによるバッファあふれ制御を行うか行わないかを指定するものです。次のいずれかを指定することができます。

Xon（XON／XOFF）	バッファあふれ制御あり（ソフト制御）
Rts（RTS／CTS）	バッファあふれ制御あり（ハード制御）
None（NONE）	バッファあふれ制御なし

２．コマンド行からの実行

コマンド行で 書式 に従ったパラメータを設定した場合、メニュー画面に入らずにコマンドを実行します。

各パラメータの間は、スペースで区切ってください。たとえば、

SPEED 9600 B8 PE S1 NONE ↵

のように入力します。各パラメータについては、「1．メニュー画面からの実行」を参照してください。

SUBST

外部コマンド

機　能　仮想ドライブの割り当ての設定・解除

書　式　SUBST〔／D｜／L〕〔〈d1：〉〈パス名1〉〕〔〈d2：〉〈パス名2〉〕

解　説　SUBSTコマンドを利用すると、パス名で指定した実ドライブのディレクトリを仮想のドライブ名で参照したり、その逆にドライブ名をパス名で参照することができます。指定するときは、どちらか一方がドライブ名でなければなりません。

SUBSTコマンドでは、以下のスイッチが使用できます。

／D

ドライブ名を指定して、その割り当てを解除します。ドライブ名を省略すると、すべての割り当てを解除します。

／L

ドライブ名を指定して、その割り当て状況を表示します。ドライブ名を省略すると、すべての割り当ての状況を表示します。

スイッチやパラメータをすべて省略して、単にSUBSTと入力するとその使い方が表示されます。

●A：¥USR¥SATO¥DOCをドライブZで参照する

```
SUBST  Z:  A:¥USR¥SATO¥DOC ↵
```

この設定でたとえば、

```
DIR  Z: ↵
```

と入力すると、ドライブZのディレクトリ（実際はA：¥USR¥SATO¥DOCのディレクトリ）情報を表示することができます。

この場合、ドライブZは仮想ドライブ（実際に物理的に接続されていないドライブ）で、まだ割り当てられていないドライブを指定します。実際に接続されているドライブ（これを仮想ドライブに対して実ドライブと呼びます）に割り当てることも可能ですが、その場合そのドライブは解除するまで参照できなくなるので、通常は仮想ドライブを指定します。また、割り当てたドライブに対して、物理的な媒体にアクセスするようなコマンド（FORMATやDISKCOPYなど）を使用することはできません。

仮想ドライブを使用するときは、CONFIG.SYSの〝LASTDRIVE〟でドライブの最大数を設定しておきます。詳しくは「第7章　システムの構築」を参照してください。

●ドライブBをA：¥BDRVで参照する

SUBST　A：¥BDRV　B：⏎

この場合ドライブBは実ドライブですが、これ以降ドライブBでは参照できず、A：¥BDRVで参照します。また、A：¥BDRVは空のディレクトリとして、前もって作成しておきます。

●割り当てを解除する場合

SUBST　／D　Z：⏎	ドライブZの割り当てを解除します。
SUBST　／D ⏎	すべての割り当てを解除します。

●割り当て状況を表示する場合

SUBST　／L　B：⏎	ドライブBの割り当て状況を表示します。
SUBST　／L ⏎	すべての割り当て状況を表示します。割り当てがわからなくなったときに使います。

スイッチ／Lをつけて実行すると、たとえば次のように表示されます。

A：　=　A：¥
B：　<　A：¥BDRV
Z：　>　A：¥USR¥SATO¥DOC

ここで、"="は割り当てを変更されていないことを、"<"は右のパスを参照したときに左のドライブを参照することを、">"は左のドライブを参照したときに右のパスを参照することを示します。

注意

仮想ドライブや仮想ディレクトリの割り当てが設定されているときに、次のコマンドを実行する場合は注意が必要です。それぞれのコマンドの **注意** を参照してください。

COPY、DEL、DIR、REN、RMDIR、VOL

SWITCH

外部コマンド

機　能

メモリスイッチの設定

書　式

SWITCH 〔RS232C=〔<パラメータ1>〕〕〔MEMORY=〔<パラメータ2>〕〕〔BOOT=〔<パラメータ3>〕〕〔EJECT=〔<パラメータ4>〕〕〔OPT2KEY=〔<パラメータ5>〕〕〔CONTRAST=〔<パラメータ6>〕〕〔KANA=〔<パラメータ7>〕〕〔TV_CTRL=〔<パラメータ8>〕〕〔LCD_MODE=〔<パラメータ9>〕〕〔SRAM=〔<パラメータ10>〕〕〔Pn=〔<パラメータ11>〕〕〔XCHG=〔<パラメータ12>〕〕〔WAIT_PRN=〔<パラメータ13>〕〕〔FIRST_KY=〔<パラメータ14>〕〕〔NEXT_KEY=〔<パラメータ15>〕〕〔ID=〔<パラメータ16>〕〕

解　説

SWITCHコマンドは、メモリスイッチ(スタティックRAM)の状態を設定するときに使用します。

1．メニュー画面からの実行

SWITCH ⏎

と入力すると、次のようなメニュー画面が表示されます。

画面の左側の欄には現在の値、右側の欄には初期値が表示されています。

この画面では、次のキー操作が可能です。

キー	機能
ESC	キーワードの欄では、SWITCHコマンドの終了。設定前の状態で終了します。パラメータ設定項目では、前に設定してあった状態に戻し、カーソルをキーワードの欄に移動します。
↵	各項目の入力になります。
↑	各項目の値をアップ
スペース	各項目の値をアップ
↓	各項目の値をダウン
↑ ↓ → ←	項目の移動

内容を変更するときは、まずキーワードの部分で↵を押します。カーソルが現在値の欄に移動しますから、そこで↑や↓で内容を変更してください。

設定が終わったら↵を押してキーワードの欄に戻り、「終了」の項目にカーソルを移動して、↵を押します。画面には確認のメッセージが表示されますので、設定した内容で更新するときは Y を押してください。なお、途中で終了するときは ESC を押します。確認のメッセージが表示されますので、任意のキーを押してください。

それぞれの項目については、次の説明を参照してください。

(1) RS-232Cインターフェイスの設定（RS232C）

RS－232Cインターフェイスのパラメータを設定します。AUX 1とは本体内蔵のRS−232Cインターフェイスです。別売のRS−232Cボードをスロットに差し込み、デバイスドライバとしてRSDRV. SYSを組み込むと、AUX 1の項目が↓↑によりAUX 2～AUX 5に変わり、それぞれのパラメータを設定することができます。

●ボーレート

データを転送する速度（bps）を指定します。

19200ボー，9600ボー，4800ボー，2400ボー，1200ボー，
600ボー，300ボー，150ボー，75ボー

のうちいずれかを指定することができます。

●データ長

データのキャラクタの長さ（ビット数）を指定します。

8ビット、7ビット、6ビット、5ビット

のうちいずれかを指定することができます。

●パリティチェック

パリティチェックのモードを指定します。

パリティ偶数、パリティ奇数、パリティなし

のうちいずれかを指定することができます。

●ストップビット数

ストップビットの数を指定します。

ストップ1、ストップ1.5、ストップ2

のうちいずれかを指定することができます。

●フロー制御の指定

XON、XOFFコードまたは、RTS、CTSによるバッファあふれ制御を行うか行わないかを指定します。次のいずれかを指定することができます。

Xon（XON／XOFF）	**バッファあふれ制御あり（ソフト制御）**
Rts（RTS／CTS）	**バッファあふれ制御あり（ハード制御）**
None（NONE）	**バッファあふれ制御なし**

(2) メモリサイズの設定（MEMORY）

メインメモリの容量を設定します。メインメモリの容量に合わせて設定してください。また、別売の増設RAMを使用する場合は、増設RAMを含めたメモリ容量を設定してください。

(3) Human68kを起動する装置の設定（BOOT）

Human68kを起動するディスクドライブやROMなどを指定します。指定できるパラメータは次の7つです。

●STD

システムで規定した順にドライブをさがして、システムディスクが挿入されているドライブからシステムを起動します。システムで規定された起動の順番は次のとおりです。

1MB フロッピーディスクドライブ0

1MB フロッピーディスクドライブ1

⋮

SCSI規格の装置でID番号が0番

SCSI規格の装置でID番号が1番

⋮

●SCSI0～SCSI7

SCSI規格の装置からシステムを起動します。SCSIの後ろにつく0～7は起動する装置のID番号を表します。CPU本体の持つSCSIコントローラにもID番号が割り当てられます。工場出荷時では、内蔵ハードディスクのID番号が0番、CPU本体のID番号が7番に割り当てられています。CPU本体のID番号はSCSI_IDの項で変更することができます。

●2HD0～2HD3

フロッピーディスクドライブからシステムを起動します。2HDの後ろにつく0～3は、起動するドライブ番号を表します。

●RAM／ROM

RAMまたはROMからシステムを起動します。

RAM	＄nnnnnn	＄nnnnnnからシステムを起動します。
RAM0	＄ED0100	＄ED0100からシステムを起動します
RAM1	＄ED0C40	＄ED0C40から 〃
ROM	＄nnnnnn	＄nnnnnnからシステムを起動します。

(4) フロッピーディスクイジェクトの設定（EJECT）

指定できるパラメータは次の2つのうちいずれかです。

●On

本体前面の電源スイッチを〝切〟(OFF) にしたとき、フロッピーディスクが自動的にイジェクトされます。

●Off

本体前面の電源スイッチを〝切〟(OFF) にしても、フロッピーディスクは自動的にイジェクトされなくなります。

(5) [OPT. 2] キーの設定（OPT 2 KEY）

[OPT.2] キーを使用して行う専用ディスプレイテレビのコントロールに関するものです。指定できるパラメータは次の2つのうちいずれかです。

●Tvctrl

[OPT.2] キーによって専用ディスプレイテレビのチャンネルを選択するなどのコントロールができるようになります。

●Normal

専用ディスプレイテレビのコントロールができなくなります。

(6) 画面のコントラストの設定（CONTRAST）

ディスプレイのコントラストを設定します。設定できるのは0〜15の値で、数値が大きいほどコントラストが強く（文字が明るく）なります。0を指定すると、まったく表示されなくなります。

(7) キーボードのかな配列の設定（KANA）

キーボードのかな配列を設定します。指定できるパラメータは次の2つのうちいずれかです。

●Jis

かな配列をキーボード上に表記されたとおりのJIS配列に設定します。

●Aiu

かな配列を50音順に設定します。

(8) 本体電源OFF時の専用ディスプレイテレビの状態の設定（TV_CTRL）

本体の電源をOFFにしたときの専用ディスプレイテレビの状態を設定します。指定できるパラメータは、次の2つおよび＄1～＄3F（＄5、＄Dを除く）の範囲の指定が可能です。

・Tv

本体の電源をOFFにしたとき、テレビ画面を表示します。

・Off

本体の電源をOFFにしたとき、専用ディスプレイテレビの電源をOFFにします。

なお、＄1～＄3Fの指定については「付録3　テレビコントロールコード一覧」を参照してください。

(9) 電卓の表示モードの設定（LCD_MODE）

[OPT.1]と[OPT.2]を同時に押すことで表示される電卓の表示モードを設定します。指定できるパラメータは次の2つのいずれかです。

●Normal

通常の文字を表示します。

●Lcd

7セグメントで文字を表示します。

(10) スタティックRAMの使用モードの設定（SRAM）

スタティックRAMの使用モードを設定します。指定できるパラメータは次の3つのいずれかです。

●No_use

スタティックRAMをSRAMディスクやプログラムエリアとして使用できません。

●Ramdisk

スタティックRAMをSRAMディスクとして使用できます。ただし、デバイスドライバとしてSRAMDISK.SYSを組み込む必要があります。

●Program

スタティックRAMをプログラムエリアとして使用できます。

⑾ テキストパレットの規定値の設定（P0～P8）

テキストパレットのシステム規定値を設定します。G（緑）、R（赤）、B（青）のそれぞれの成分について、＄1～＄1Fまでの各色32階調の範囲で設定できます。

⑿ "¥"、"￣"、"|" キーの表示モードの設定（XCHG）

"¥"、"￣"、"|" キーの表示モードを設定します。指定できるパラメータは次の0～7までの値です。

0…"¥" キーを押すと "¥" を、"￣" を押すと "￣" を、"|" を押すと "|" を表示します。

1…"¥" キーを押すと "＼" を、"￣" を押すと "￣" を、"|" を押すと "|" を表示します。

2…"¥" キーを押すと "¥" を、"￣" を押すと "～" を、"|" を押すと "|" を表示します。

3…"¥" キーを押すと "＼" を、"￣" を押すと "～" を、"|" を押すと "|" を表示します。

4…"¥" キーを押すと "¥" を、"￣" を押すと "￣" を、"|" を押すと "¦" を表示します。

5…"¥" キーを押すと "＼" を、"￣" を押すと "￣" を、"|" を押すと "¦" を表示します。

6…"¥" キーを押すと "¥" を、"￣" を押すと "～" を、"|" を押すと "¦" を表示します。

7…"¥" キーを押すと "＼" を、"￣" を押すと "～" を、"|" を押すと "¦" を表示します。

⒀ プリンタのタイムアウト時間の設定（WAIT_PRN)

プリンタのタイムアウト時間を設定します。約5秒から5秒刻みで設定できます。プリンタのタイムアウト時間とは、プリンタがプリント開始可能信号(Busy信号がLow)を返すまでのコンピュータの待機時間です。

⒁ キーのオートリピート間隔の設定（FIRST_KY)

キーのオートリピートが始まるまでの時間を設定します。0～15の範囲で指定できます。指定した数値の右に実際の時間が表示されます。

⒂ キーのリピート間隔の設定（NEXT_KEY）

キーのリピート間隔を設定します。０～15の範囲で指定できます。
指定した数値の右に実際の時間が表示されます。

⒃ CPU本体のID番号の設定（SCSI_ID）

CPU本体のID番号を設定します。
０～７の範囲の範囲で指定できます。
工場出荷時では、CPU本体のID番号が７番に割り当てられています。内蔵ハードディスクや接続しているSCSI装置のID番号と重複しないように設定してください。

２．コマンド行からの実行

コマンド行で 書式 に従ったパラメータを設定した場合、メニュー画面に入らずにコマンドを実行します。
指定できるパラメータは次のとおりです。なお、〔 〕で示したものは省略形です。

⑴ RS－232Cインターフェイスの設定（RS232C）

RS232C＝<パラメータ１>

RS－232Cインターフェイスのパラメータを設定します。キーワードは"R＝"と略すこともできます。
設定できるパラメータは次の５つです。パラメータを複数指定するときは、スペースで区切って指定してください。

●ボーレート
データを転送する速度（bps）を指定します。

19200,9600,4800,2400,1200,600,300,150,75

のうちいずれかを指定することができます。

●データ長
データのキャラクタの長さ（ビット数）を指定します。

BITS－８〔B８〕、BITS－７〔B７〕、BITS－６〔B６〕、BITS－５〔B５〕

のうちいずれかを指定することができます。

●パリティチェック

パリティチェックのモードを指定します。

PARITY-NONE〔PN〕、PARITY-EVEN〔PE〕、PARITY-ODD〔PO〕

のうちいずれかを指定することができます。

●ストップビット数

ストップビットの数を指定します。

STOP－1〔S1〕、STOP－1.5〔S1.5〕、STOP－2〔S2〕

のうちいずれかを指定することができます。

●フロー制御の指定

XON、XOFFコードまたは、RTS、CTSによるフロー制御を行うか行わないかを指定します。次のいずれかを指定することができます。

Xon (XON／XOFF)	バッファあふれ制御あり（ソフト制御）
Rts (RTS／CTS)	バッファあふれ制御あり（ハード制御）
None (NONE)	バッファあふれ制御なし

(2) メモリサイズの設定（MEMORY）

MEMORY＝<パラメータ2>

メインメモリの容量を設定します。メインメモリの容量に合わせて設定してください。キーワードは〝M=″と略すこともできます。

(3) Human68kを起動する装置の設定（BOOT）

BOOT＝<パラメータ3>

Human68kを起動するディスクドライブやROMなどを指定します。キーワードは〝B=″と略すこともできます。

指定できるパラメータは次の7つです。

●STD

システムで規定した順にドライブをさがして、システムディスクが挿入されているドライブからシステムを起動します。システムで規定された起動の順番は次のとおりです。

1MB フロッピーディスクドライブ 0
1MB フロッピーディスクドライブ 1
⋮
SCSI規格の装置でID番号が 0 番
SCSI規格の装置でID番号が 1 番
⋮

●SCSI0～SCSI7

SCSI規格の装置からシステムを起動します。SCSIの後ろにつく0～7は起動する装置のID番号を表します。CPU本体の持つSCSIコントローラにもID番号が割り当てられます。工場出荷時では、内蔵ハードディスクのID番号が0番、CPU本体のID番号が7番に割り当てられます。CPU本体のID番号はSCSI_IDの項で変更することができます。

●2HD0～2HD3

フロッピーディスクドライブからシステムを起動します。2HDの後ろにつく 0 ～ 3 は、起動するドライブ番号を表します。

●RAM／ROM

RAMまたはROMからシステムを起動します。

RAM	＄nnnnnn	＄nnnnnnからシステムを起動します。
RAM0	＄ED0100	＄ED0100からシステムを起動します
RAM1	＄ED0C40	＄ED0C40から 〃
ROM	＄nnnnnn	＄nnnnnnからシステムを起動します。

(4) フロッピーディスクイジェクトの設定（EJECT）

EJECT=＜パラメータ4＞

キーワードは“E=”と略すこともできます。
指定できるパラメータは次の2つのうちいずれかです。

●ON
本体前面の電源スイッチを“切”(OFF)にしたとき、フロッピーディスクは自動的にイジェクトされます。

●OFF
本体前面の電源スイッチを“切”(OFF)にしても、フロッピーディスクは自動的にイジェクトされなくなります。

(5) [OPT.2] キーの設定（OPT2KEY）

OPT2KEY=＜パラメータ5＞

[OPT.2] キーを使用して行う専用ディスプレイテレビのコントロールに関するものです。キーワードは“O=”と略すこともできます。
指定できるパラメータは次の2つのうちいずれかです。

●TVCTRL
[OPT.2] キーによって専用ディスプレイテレビのチャンネルを選択するなどのコントロールができるようになります。

●NORMAL
専用ディスプレイテレビのコントロールができなくなります。

(6) 画面のコントラストの設定（CONTRAST）

CONTRAST=＜パラメータ6＞

ディスプレイのコントラストを設定します。キーワードは“C=”と略すこともできます。
設定できるのは0～15の値で、数値が大きいほどコントラストが強く（文字が明るく）なります。0を指定すると、まったく表示されなくなります。

(7) キーボードのかな配列の設定（KANA）

KANA=＜パラメータ7＞

キーボードのかな配列を設定します。キーワードは"K="と略すこともできます。指定できるパラメータは次の2つのうちいずれかです。

●JIS
かな配列をキーボードに表記されたとおりのJIS配列に設定します。

●AIU
かな配列を50音順に設定します。

(8) 本体電源OFF時の専用ディスプレイテレビの状態の設定（TV_CTRL）

TV_CTRL=＜パラメータ8＞

本体の電源をOFFにしたときの専用ディスプレイテレビの状態を設定します。指定できるパラメータは、次の2つおよび$1～$3F（$5、$Dを除く）の範囲の指定が可能です。

●TV
本体の電源をOFFにしたとき、テレビ画面を表示します。
●OFF
本体の電源をOFFにしたとき、専用ディスプレイテレビの電源をOFFにします。

なお、$1～$3Fの指定については「付録3　テレビコントロールコード一覧」を参照してください。

(9) 電卓の表示モードの設定（LCD_MODE）

LCD_MODE=＜パラメータ9＞

OPT.1 と OPT.2 を同時に押すことで表示される電卓の表示モードを設定します。キーワードは"L="と略すこともできます。
指定できるパラメータは次の2つのいずれかです。

●NORMAL
通常の文字を表示します。

●LCD
7セグメントで文字を表示します。

⑽ スタティックRAMの使用モードの設定（SRAM）

SRAM=＜パラメータ10＞

スタティックRAMの使用モードを設定します。キーワードは〝S=〟と略すこともできます。

指定できるパラメータは次の3つのいずれかです。

●N

スタティックRAMをSRAMディスクやプログラムエリアとして使用できません。

●R

スタティックRAMをSRAMディスクとして使用できます。ただし、デバイスドライバとしてSRAMDISK.SYSを組み込む必要があります。

●P

スタティックRAMをプログラムエリアとして使用できます。

⑾ テキストパレットの規定値の設定（P0～P8）

Pn=＜パラメータ11＞

テキストパレットのシステム規定値を設定します。$0～$FFFFの範囲で設定します。詳しくは「メニュー画面」を参照してください。

⑿ 〝¥〟、〝‾〟、〝|〟キーの表示モードの設定（XCHG）

XCHG=＜パラメータ12＞

〝¥〟、〝‾〟、〝|〟キーの表示モードを設定します。キーワードは〝X=〟と略すこともできます。

指定できるパラメータは次の0～7までの値です。

0…〝¥〟キーを押すと〝¥〟を、〝‾〟を押すと〝‾〟を、〝|〟を押すと〝|〟を表示します。
1…〝¥〟キーを押すと〝＼〟を、〝‾〟を押すと〝‾〟を、〝|〟を押すと〝|〟を表示します。
2…〝¥〟キーを押すと〝¥〟を、〝‾〟を押すと〝～〟を、〝|〟を押すと〝|〟を表示します。
3…〝¥〟キーを押すと〝＼〟を、〝‾〟を押すと〝～〟を、〝|〟を押すと〝|〟を表示します。

4… "¥" キーを押すと "¥" を、"￣" を押すと "￣" を、"|" を押すと "¦"
を表示します。
5… "¥" キーを押すと "＼" を、"￣" を押すと "￣" を、"|" を押すと "¦"
を表示します。
6… "¥" キーを押すと "¥" を、"￣" を押すと "～" を、"|" を押すと "¦"
を表示します。
7… "¥" キーを押すと "＼" を、"￣" を押すと "～" を、"|" を押すと "¦"
を表示します。

⒀ プリンタのタイムアウト時間の設定（WAIT_PRN）

WAIT_PRN=<パラメータ13>

プリンタのタイムアウト時間を設定します。0～＄FFFFFFFFの範囲で設定できます。＄80000＝約5秒を目安としてください。キーワードは"W="と略すこともできます。プリンタのタイムアウト時間とは、プリンタがプリント開始可能信号(Busy信号がLow）を返すまでのコンピュータの待機時間です。

⒁ キーのオートリピート間隔の設定（FIRST_KY）

FIRST_KY=<パラメータ14>

キーのオートリピートが始まるまでの時間を設定します。n＝0～15の範囲で指定できます。このnの示す時間は、200+100×n（単位ms）です。キーワードは"F="と略すこともできます。

⒂ キーのリピート間隔の設定（NEXT_KEY）

NEXT_KEY=<パラメータ15>

キーのリピート間隔を設定します。n=0～15の範囲で指定できます。
このnの示す時間は、$30+5\times n^2$（単位ms）です。キーワードは"N="と略すこともできます。

⒃　CPU本体のID番号の設定（SCSI_ID）

ID=<パラメータ16>

CPU本体のID番号を設定します。n＝0～7の範囲で指定できます。

工場出荷時ではCPU本体のID番号が7番に割り当てられています。内蔵ハードディスクや接続しているSCSI装置のID番号と重複しないように設定してください。

SYS

外部コマンド

機　能　Human68k システムの転送

書　式　SYS　〈d：〉

解　説

SYS コマンドは、カレントドライブ中にある Human68k のシステムファイルを、〈d：〉で指定したドライブにコピーします。

SYS コマンドによってシステムファイルをコピーしたディスクは、Human68k を起動することのできるシステムディスクとなります。

転送先のドライブ名〈d：〉は、必ず指定してください。

システムファイルのコピー先となるディスクは、事前に FORMAT コマンドでフォーマットされたものでなくてはならず、また次のいずれかを満たしていなくてはなりません。

●ファイルやディレクトリが何も作られていないブランクディスク

フォーマット直後か、すべてのファイルおよびディレクトリを削除したディスクを指します。

●すでにシステムファイルが存在しているディスク

以前に、システムファイルを転送したことのあるディスクです。これは、システムのバージョンが変わったときに、システムを入れかえる必要がある場合を指します。

カレントドライブのディスクにシステムファイルがあり、それをドライブ B のディスクにコピーする場合は、

```
SYS  B: ↵
```

とすればコマンドが実行され、システムファイルが転送されます。

ここで、転送の対象となるシステムファイルは、ファイルの属性として "S" (システム)をもっているものに限られます。ファイルの属性についての詳細は、ATTRIB コマンドの項を参照してください。

転送が成功すると、次のようにメッセージが表示されます。

```
HUMAN. SYS
      ・
      ・
      ・
システムの転送を終了しました
```

システムの転送中、次のようなメッセージが表示されることがあります。

XXXXXX．XXX は連続した FAT に転送されていません

ここで、XXXXXX．XXX は、ファイル名です。

もし、このメッセージが HUMAN．SYS ファイルについて出されたときには、そのままではそのディスクをシステムディスクとして使うことはできません。そのディスクの中のすべてのファイルおよびディレクトリを削除するか、またはフォーマットし、その後にもう一度 SYS コマンドを実行し直してください。

他のシステムファイルについては、このメッセージが表示されても問題はありません。

TEMP

内部コマンド

機能 テンポラリファイルのパスの設定

書式 TEMP 〔[<d：>] <パス名>〕

解説 テンポラリファイルのパスを設定します。Human68k のパイプ機能やアセンブラ、コンパイラなどを使用すると、処理中に自動的に中間ファイル(これをテンポラリファイルと呼びます)を作成します。処理時間の多くがこのファイルの入出力に使われるので、できるだけ高速なディスクに対して TEMP コマンドでパスを設定するとそれだけ処理も速くなります。フロッピーディスクよりハードディスク、ハードディスクより RAM ディスクが高速なので、パスは RAM ディスクに設定するともっとも効果的です。

RAMディスクは、デバイスドライバとしてRAMDISK.SYSを組み込みます。詳しくは「第7章　システムの構築」を参照してください。

●テンポラリファイルのパスをA：¥TMPに設定する

```
TEMP　A：¥TMP　⏎
```

●テンポラリパスを表示する

```
TEMP　⏎
```

パス名を設定しない場合は、現在の設定を表示します。

TIME

内部コマンド

機　能　時刻の表示・設定

書　式　TIME　〔〈hh〉：〈mm〉：〈ss〉〕

解　説

Human68kは、クロック(時計)により、日付・時刻を管理しています(電源を切っても常に日付と時刻は保持されています)。これらの時間情報は、ファイルを作成・更新したときにディレクトリに記録されます。

TIMEコマンドは、これらの時間情報のうちの〝時刻〞を表示したり、設定したりするものです。

```
TIME ↵
```

と、コマンドだけを入力すると、

```
現在の時刻は〈hh〉：〈mm〉：〈ss〉です
時刻を入力してください：
```

と表示されます。表示された時刻を変更しないときはそのままリターンキーを押します。新たに時刻を設定するときは、次のような書式で正しく入力します。

```
〈hh〉：〈mm〉：〈ss〉 ↵
```

時〈hh〉、分〈mm〉、秒〈ss〉のいずれも、省略することはできません。それぞれの設定可能な範囲は以下のとおりです。

```
時：〈hh〉　＝00～23
分：〈mm〉　＝00～59
秒：〈ss〉　＝00～59
```

時刻の設定が正しくないと、

```
時刻の指定が違います
時刻を入力してください：
```

というエラーメッセージが表示されます。もう一度正しく入力してください。

なお、TIMEコマンドをコマンドモードから入力する際に、直接時刻のパラメータを付けて実行することもできます。

```
TIME　10：10：30 ↵
```

このとき、時刻が正しければメッセージは何も表示されず、そのまま時刻が設定されます。時刻が誤っていると、前述と同じエラーメッセージが表示されますので、再入力してください。

TIMER

外部コマンド

機　能

時刻の表示・アラームのセット

書　式

(1)　TIMER ｛／ON〔〈番号〉〕｜／OFF｝
(2)　TIMER　／A　〈時刻〉　〈ファイル名〉
(3)　TIMER　／D　〈時刻〉　〈ファイル名〉
(4)　TIMER　／T　〈時刻〉　〈テレビ制御文字〉
(5)　TIMER　／L
(6)　TIMER　／K　〈番号〉

解　説

TIMERコマンドは並行処理用のコマンドとして起動され、時刻の表示や時刻指定によってADPCMの再生、メッセージの表示、テレビのコントロールなどができます。時刻指定は最大20まで可能です。

TIMERコマンドを実行するには、あらかじめCONFIG.SYSにPROCESSの項を追加する必要があります。追加するには、スクリーンエディタEDを使って、たとえば次のようにします。

Human68kのコマンドモードのコマンド入力待ち状態で、

```
ED  CONFIG.SYS  ⏎
```

と入力すると、CONFIG.SYSファイルの内容が表示されます。ここで、最下行に次の行を追加します。

```
PROCESS = 10  2  20
```

この作業が終ったあと、[ESC]・[E]（[ESC]を押し、続けて[E]を押す）を押して、変更したCONFIG.SYSをディスクにセーブして終了します。

CONFIG.SYSが書き替えられましたので、本機のリセットスイッチを押して再起動してください。これで、TIMERコマンドが実行できるようになりました。このあと、実際にTIMERコマンド（書式の(2)〜(6)）を実行する前に、必ず

```
TIMER  /ON  〔〈番号〉〕 ⏎
```

または

```
TIMER  /OFF  ⏎
```

を実行してください。

CONFIG.SYSのPROCESSについては、第1部「7.3 CONFIG.SYSファイルのコマンド」をご覧ください。

TIMERコマンドでは、以下のスイッチが使用できます。

／ON〔〈番号〉〕

時刻を画面のファンクションキーの上に表示します。〈番号〉で時刻を表示するファンクションキーの番号を指定します。番号の指定がないときは、F10の位置に表示されます。画面が96桁のときは〔時：分：秒〕、64桁のときは〔時：分〕で表示します。

／OFF

時刻を表示しないようにします。

／A　〈時刻〉　〈ファイル名〉

〈時刻〉で指定された時刻に〈ファイル名〉で指定したファイルをADPCMのデータファイルとして再生します。再生できるのは15.6kHzでサンプリングされたもので、$FF00バイト以下の長さのファイルです。これ以上長い場合は無視されます。

たとえば、10時15分にビープ音を鳴らすときは、

```
TIMER ／A 10:15 ¥CONFIG¥BEEP.SYS ↵
```

と入力します。

／D　〈時刻〉　〈ファイル名〉

〈時刻〉で指定された時刻に〈ファイル名〉で指定したファイルをウィンドウに約10秒間表示します。ファイルはテキストファイルで60桁×25文字まで表示可能です。

／T　〈時刻〉　〈テレビ制御文字〉

〈時刻〉で指定された時刻に〈テレビ制御文字〉で指定したテレビのコントロールを行います。〈テレビ制御文字〉には次のような指定ができます。

OFF	専用ディスプレイテレビの電源をOFF
CH1〜CH12	専用ディスプレイテレビの電源をONにしてチャンネルを1〜12にセット
TV	専用ディスプレイテレビの電源をONにしてテレビ画面にする
COM	専用ディスプレイテレビの電源をONにしてコンピュータ画面にする
SP1	専用ディスプレイテレビの電源をONにしてスーパーインポーズ（コントラストダウン）画面に設定する
SP2	専用ディスプレイテレビの電源をONにしてスーパーインポーズ（コントラストノーマル）画面に設定する

／L

設定されているタイマーの一覧を表示します。

／K 〈番号〉

〈番号〉には一覧表示で表示される管理番号を指定して、そのアラームを解除します。〈番号〉を省略すると、すべてを解除します。

スイッチを省略して、単に〝TIMER〟とだけ入力すると、TIMER コマンドの使い方が表示されます。

●現在の時刻をファンクションキー１の上に表示する

TIMER ／ON 1 ⏎

●12時00分にA：¥PCM¥OHIRU.PCM を再生する

TIMER ／A 12：00 A：¥PCM¥OHIRU.PCM ⏎

●19時30分に専用ディスプレイテレビの電源を ON にしてチャンネルを３にする

TIMER ／T 19：30 CH 3 ⏎

●9時00分に B：¥MEMO.DOC を表示する

TIMER ／D 9：00 B：¥MEMO.DOC ⏎

●タイマーの一覧を表示する

TIMER ／L ⏎

これらを実行すると、たとえば次のように表示されます。

1）A 12：00 A：¥PCM¥OHIRU.PCM
2）T 19：30 CH 3
3）D（9：00）B：¥MEMO.DOC

ここで、先頭の番号は管理番号で１～20までの値です。また、（9：00）のようにカッコで表示されたものはすでに処理済みであることを示します。

●登録されているアラームを解除する（管理番号の２を解除する）

TIMER ／K 2

TREE

外部コマンド

機能　ディレクトリ構造のツリー形式での表示

書式　TREE 〔/F〕〈パス名〉

解説　〈パス名〉が省略された場合は、カレントドライブのルートディレクトリ以下のディレクトリ構造がツリー形式で表示されます。

TREE ↵

カレントドライブの、ルートディレクトリ以下のディレクトリ構造がツリー形式で表示されます。

〈パス名〉で指定したディレクトリ以下のディレクトリ構造を、ツリー形式で表示します。

TREE ¥TMP ↵

カレントドライブの、ディレクトリ¥TMP以下のディレクトリ構造がツリー形式で表示されます。

●/F

ディレクトリ構造をツリー形式で表示するとともに、そのディレクトリに属するファイルの名前もすべて表示します。

TREE /F ↵

カレントドライブの、ルートディレクトリ以下のディレクトリ構造がツリー形式で表示されます。ディレクトリに属するファイルの名前もすべて表示します。

なお、システム属性、不可視属性のついたディレクトリ、ファイルもすべて表示します。

```
¥
|-HUMAN.SYS
|-HIT.BAT
|-MANA.BAT
|-JUN.BAT
|-SANA.BAT
|-MAG.BAT
|-G3.BAT
|
|-DOC
|     |-HOSTS.EQU
|     |-HOSTS.LPD
|     |-INETD.CON
|     |-NETWORKS
|     |-MANA.DOC
|     |-JUN.DOC
|     |-SANA.DOC
```

TYPE

内部コマンド

機　能	ファイル内容の表示
書　式	TYPE　〈ファイル名〉〔〈ファイル名〉・・・〕

解　説

TYPEコマンドは、ファイルの内容を表示するときに使用します。

〈ファイル名〉に、表示したいファイルを指定します。複数個指定すると、指定した順にファイルの内容が表示されます。ワイルドカードを用いることもできます。

TYPEコマンドでは、テキストファイルの内容は表示することができますが、バイナリファイルを指定したときはコントロール文字も記号として表示されてしまうため、正しい表示が行われません。バイナリファイルの内容を表示するにはDUMPコマンドを使用してください。

USKCGM

外部コマンド

機　能　　外字の作成・登録・削除

書　式　　USKCGM

解　説

USKCGM コマンドは、外字ファイルの作成、外字ファイルのシステムへの登録、外字ファイルからの指定外字の削除などを行います。

USKCGM ⏎

と実行すると、

更新ですか、登録ですか？ <U／L>

と表示されます。

1．外字ファイルの登録

外字ファイルの内容をシステムに登録するときは、

L

を押します。⏎(リターンキー)を押す必要はありません。次に、

入力ファイル名〔USKCG．SYS〕:

と表示されますから、登録したい外字ファイルのファイル名を入力します。そのまま⏎(リターンキー)を押すと USKCG．SYS ファイルが登録されます。

外字ファイルがシステムに登録されると、

ファイルの内容をシステムに登録しました

と表示され、コマンドモードに戻ります。

2．外字ファイルの更新

外字ファイルを更新するときは、

U

を押します。⏎(リターンキー)を押す必要はありません。次に、

入力ファイル名〔USKCG．SYS〕:

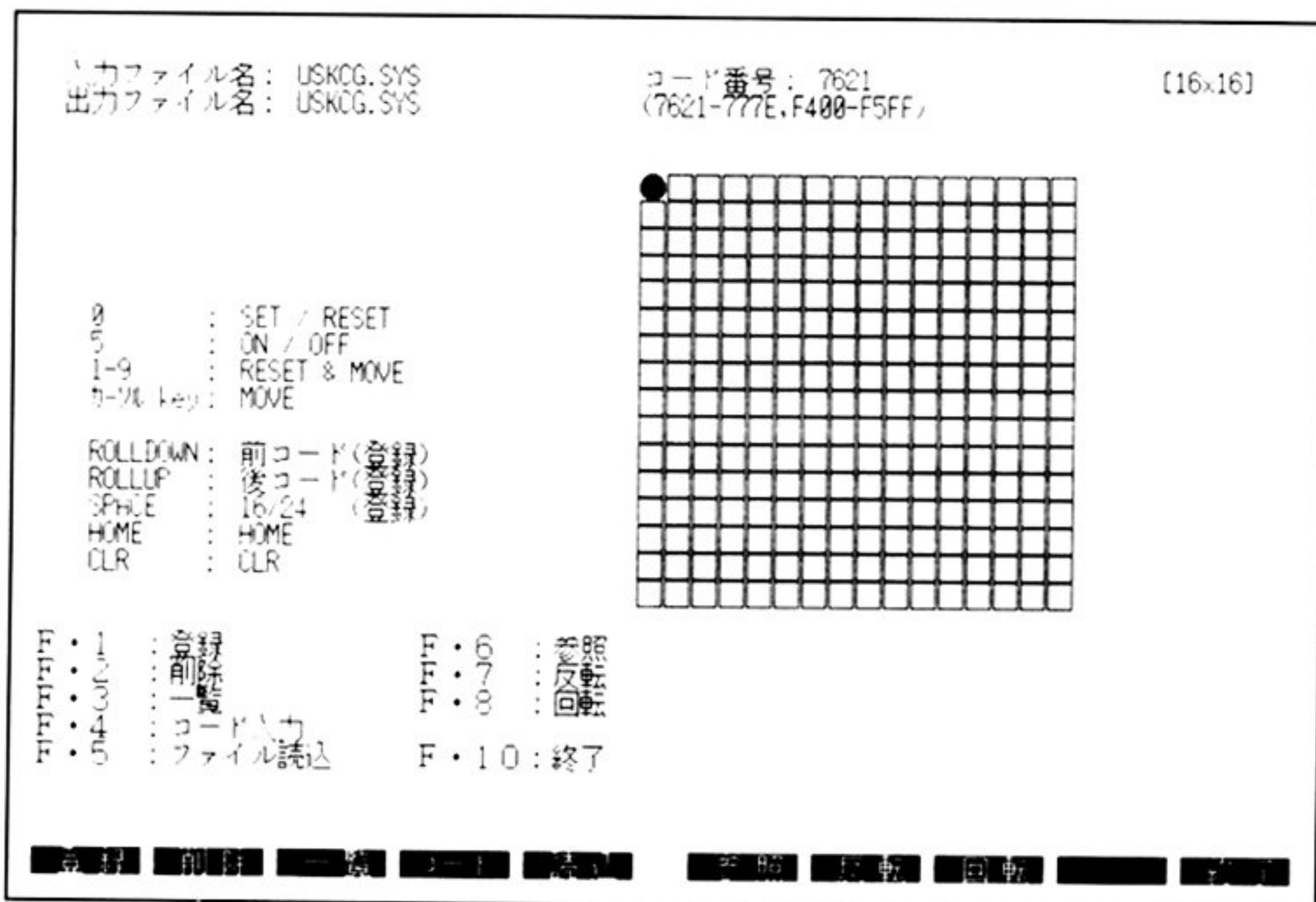
と表示されますから、読み込みたい外字データの格納された、外字ファイルのファイ

ル名を入力します。そのまま ⏎（リターンキー）を押すと USKCG.SYS ファイルが指定されます。

次に、

出力ファイル名〔USKCG. SYS〕:

と表示されますから、同様にして、作成した外字データを書き込みたい外字ファイルのファイル名を入力します。⏎（リターンキー）のみ押せば、USKCG.SYS ファイルが指定されます。

すると、次のような画面が表示されます。

次に各キーの使用法を解説します。

ROLLDOWN : 前コード

ROLL DOWN キー。１つ前の画面の編集に移ります。

ROLLUP : 後コード

ROLL UP キー。１つ後の外字の編集に移ります。

SPACE : 16／24

スペースキー。パターンのドット数(16×16ドット、24×24ドット)の切りかえを行います。

HOME : HOME

HOME キー。編集画面のホームポジション(左上)にカーソルを移動します。

CLR：CLR

CLR キー。編集画面をクリアします。

0：SET／RESET

数値入力キーの 0 。1～9でカーソルを移動するときに、白で埋めながら(セット)移動するか、黒で埋めながら（リセット）移動するかを決定をします。

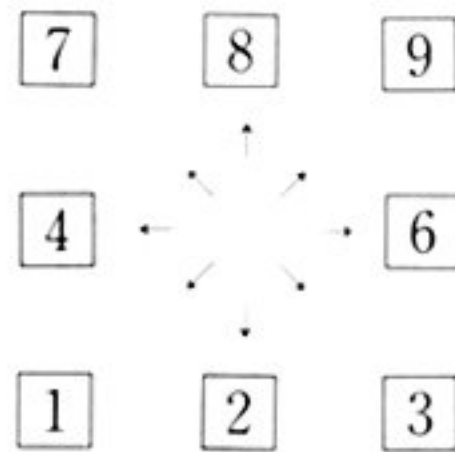
5：ON／OFF

数値入力キーの 5 。カーソルⅠの白黒を反転します。カーソルキーでカーソルを移動する場合に使用します。

1～9：SET & MOVE

1～9：RESET & MOVE

数値入力キーの 1 ～ 9 。0（SET／RESET）で決めた状態に従って、カーソルをセット移動またはリセット移動します

各キーでのカーソルの動きは、次のとおりです。

カーソル key：MOVE

カーソルを移動します。セットもリセットもしません。

F1：登録

新しく作成したパターンを外字データとしてシステムに書き込みます。

画面上部に表示されているコード番号で書き込みが行われ、画面上では何も起こりません。

F2：削除

システムに格納されている外字を削除します。

画面左部に、角形カッコ〔　〕が表示され、コード番号の入力を求めてきます。16進数で削除したい外字のコード番号を入力してください。

F3：一覧

システムに格納されている外字の一覧を表示します。

編集画面はクリアされ、外字の一覧が表示されます。次の画面を見たいときは任意のキー、前の画面を見たいときは ROLL DOWN キー、一覧表示を中止したいときは ESC キーをそれぞれ押してください。

F４：コード

指定したコード番号の外字の編集に移ります。

画面左部に、角形カッコ〔　〕が表示され、コード番号の入力を求めてきます。16進数で編集したい外字のコード番号を入力してください。

F５：読込

外字ファイルから新しく外字データを読み込みます。

画面左部に、次のようにメッセージが表示されます。

入力ファイル名〔USKCG．SYS〕：

読み込みたい外字データの格納された、外字ファイルのファイル名を入力します。そのまま[↵](リターンキー)を押すとUSKCG．SYSファイルが指定されます。

[F５]によって読み込みを行うと、それまでに編集していた外字データはすべて消去され、新しく読み直されます。注意してください。

F６：参照

外字の編集中に、システムに登録されている文字パターンを呼び出します。

画面左部に、角形カッコ〔　〕が表示され、コード番号の入力を求めてきます。16進数で呼び出したい文字のコード番号を入力してください。

外字のコード番号を入力した場合は、外字ファイルからではなく、現在システムに登録されている外字のパターンが呼び出されます。

編集画面には、このキーによって呼び出された文字パターンが表示されます。このとき画面上部に示されたコード番号は変わりません。呼び出したパターンを再編集し、さきに編集していたコード番号の外字として登録することができます。

文字のコード番号については、『日本語入力・辞書ユーティリティユーザーズマニュアル』の「付録　3　コード表」を参照してください。

F７：反転

編集画面のパターン全体を白黒反転します。

F８：回転

編集画面のパターン全体を、90°ずつ右回りに回転します。

F10：終了

外字の編集を終了します。[F10]を押すと、

処理を終了しますか？〈Y／N〉

と表示されます。ここで[N]を押すと、もとの編集画面に戻ります。[Y]を押すと、

システムに登録しました
ファイルを更新しますか？〈Y／N〉

と表示されます。ここで[Y]を押すと、編集した外字データは指定出力ファイルに書き込まれ、システムにも登録されます。そして、

ファイルの内容をシステムに登録しました

と表示され、コマンドモードに戻ります。

[N]を押すと、何もせずにそのままコマンドモードに戻ります。この場合、さきに行った編集作業はすべて無効となり、出力ファイルには何も書き込まれません。システムにも、外字は登録されません。

３．外字ファイルの内容

外字ファイルの内容は、次のようになっています。

予約領域（34バイト）	
コード	16×16のフォントデータ
コード	16×16のフォントデータ
⋮	⋮
$FFFF	
コード	24×24のフォントデータ
コード	24×24のフォントデータ
⋮	⋮

$FFFF以降は、24×24ドットのデータです。

ここで、コード、コードの内容、フォントデータの長さは、次のようになっています。

コード（２バイト）	コードの内容	フォントデータの長さ	
		16×16ドット	24×24ドット
$0000～7FFF	上位・下位が逆のJIS漢字コード	32バイト	72バイト
$8000～EFFF	シフトJIS漢字コード	32バイト	72バイト
$F400～F5FF	半角外字コード	16バイト	36バイト

注意：＄F400～＄F41F、＄F500～＄F51Fは特殊半角外字領域ですので、通常のアプリケーションソフトでは使用できません。

VER

内部コマンド

機能　システムのバージョンの表示

書式　VER

解説　VER コマンドは、Human68k のバージョンを確認するときに使います。

　　VER ↵

とすると、

　　Human68k　version　＊.＊＊

のように Human68k のバージョンが表示されます。

VERIFY

内部コマンド

機　能　ベリファイ機能の設定

書　式　VERIFY 〔ON | OFF〕

解　説　VERIFY コマンドは、ディスクに書き込みを行う際に、正常に書き込めたかどうかのチェックを行うか否かの設定をします。

VERIFY を ON に設定すると、書き込みの直後に、書き込んだデータを読み出してそのデータが正しく書けたかどうかの照合チェックを行います。したがって、不良セクタなどのために物理的に書き込めない部分があるときは、即座にエラーとして表示されるようになります。

VERIFY コマンドを使うと、照合チェックを行いながら書き込むため、ディスクに書き込みを行う速度は遅くなります。

現在の VERIFY の状態を見たいときは、

```
VERIFY ↵
```

とします。VERIFY が OFF であれば、

```
verify は<off>です
```

と表示され、ON であれば、

```
verify は<on>です
```

と表示されます。

なお、ベリファイ機能の設定は、CONFIG.SYS ファイルのコマンド VERIFY でも設定することができます。これについては、「第7章　システムの構築」を参照してください。

VOL

内部コマンド

機能　ディスクのボリュームラベルの表示・変更

書式　VOL 〔<d：>〕〔<ボリュームラベル>／S ｜ ／C〕

解説

VOL コマンドは、指定したドライブ上にあるディスクのボリュームラベルを表示したり、変更したりするときに使います。

VOL　A：↵

とすると、

ドライブ A：のボリュームラベルは XXX です

と表示されます。ここで、XXX は、ディスクのフォーマット時につけたボリュームラベルを表します。

ディスクのボリュームラベルが設定されていないときは、次のメッセージが表示されます。

ドライブ A：のボリュームラベルはありません

ドライブの指定がないときはカレントドライブが表示の対象になります。

また、<ボリュームラベル>／S をつけて実行することにより、ボリュームラベルを変更することができます。

たとえば、

VOL　A：MYDISK／S ↵

と入力すれば、ボリュームラベルが、〝MYDISK〟に変更されます。

VOL コマンドに／C をつけて実行すると、ボリュームラベルを削除することができます。

たとえば、

VOL　B：　／C ↵

と入力すれば、ドライブ B のボリュームラベルが削除されます。

注意

ボリュームラベルの設定のとき、仮想ドライブなどに設定を行ったときには、次のことに注意してください。

実体となるディレクトリにも同じボリュームラベルが設定されます。したがって、そのディレクトリの内容が空でもRMDIRコマンドでディレクトリを削除できないことになります。RMDIRコマンドでそのディレクトリを削除しようとすると、

ディレクトリが空でないか、見つからないか、カレントパスです

と表示され、指定したディレクトリを削除できません。

実体となるディレクトリを削除するときには、まず仮想ドライブが設定されているドライブのボリュームラベルを削除し、次に仮想ドライブの設定を解除してください。

また、複数の仮想ドライブが同じディレクトリに設定されているときに、どれか1つの仮想ドライブにボリュームラベルの操作を行うと、すべての仮想ドライブのボリュームラベルに対して操作したことになります。もちろん、実体となるディレクトリがルートディレクトリのときには、実体となるドライブのボリュームラベルも操作されることになります（仮想ドライブや仮想ディレクトリについては、SUBSTコマンドをご覧ください）。

WHERE

外部コマンド

機　能　ファイルの検索

書　式　WHERE 〔／A〕〔／F〕{〈d：〉|〈パス名〉|〈ファイル名〉}

解　説

〈パス名〉または〈ファイル名〉で指定したファイルを、ディレクトリのツリー構造の中から検索します。

〈d：〉で指定したドライブの〈パス名〉で指定したディレクトリから始め、〈ファイル名〉で指定されたファイルをディレクトリのツリー構造に従って順に検索し、見つかったファイルの位置するディレクトリのパス名を表示します。

〈d：〉のドライブ名が省略された場合にはカレントドライブが、〈パス名〉が省略された場合にはルートディレクトリが指定されたものと見なされます。

〈ファイル名〉にはワイルドカードが使用できます。

また、システム属性、不可視属性のついたディレクトリ、ファイルもすべて検索します。

```
WHERE  TEST.TXT ↵
```

カレントドライブのルートディレクトリから、ファイル TEST. TXT を検索し、その位置をパス名で表示します。

●／A

すべてのドライブを検索します。

```
WHERE  /A  TEST.  TXT ↵
```

すべてのドライブのルートディレクトリから、ファイル TEST.TXT を検索し、その位置をパス名で表示します。

注意：スイッチ／A をつけるときは、ドライブ名の〈d：〉は指定しないでください。

●／F

パス名のほかに、見つかったファイルの名前も表示します。

```
WHERE  /F  *.  TXT ↵
```

カレントドライブのルートディレクトリから、拡張子が〝. TXT〞であるファイルを検索し、その位置をパス名とファイル名で表示します。

5.3　バッチ処理コマンド

ここで解説するコマンドは、主にHuman68kのバッチ処理をコントロールするためのものです。これらのコマンドは単独でも実行することができますが、バッチ処理の中でその機能をフルに発揮します。バッチ処理コマンドには次のものがあります。

ECHO	バッチ処理中におけるコマンド行表示の設定・メッセージの表示
FOR	コマンドの反復実行
GOTO	バッチ処理の流れの変更
IF	条件設定によるバッチ処理
PAUSE	バッチ処理の一時停止
REM	バッチ処理中におけるコメントの表示
SHIFT	バッチ処理中におけるパラメータのシフト

ECHO

内部コマンド

機　能	バッチ処理中におけるコマンド行表示の設定・メッセージの表示
書　式	ECHO〔ON｜OFF｜〈メッセージ〉〕

解　説

ECHOコマンドは、バッチプログラム実行中にコマンド行を表示するかしないかを設定するために使用します。

また、任意のメッセージを表示するためにも使うことができます。

● ECHOモードをONにします

```
ECHO　ON ↵
```

と入力します。

この状態のときバッチファイルが起動されると、バッチファイルの内容が1行ずつ、実際の実行に合わせて画面にエコーバックされます。

● ECHOモードをOFFにします

```
ECHO　OFF ↵
```

と入力します。

この状態のときバッチファイルが起動されると、バッチファイルの内容は画面に表示されません。バッチファイル中のコマンド行を表示したくないときには、このコマンドをバッチファイルの第1行として入れておくとよいでしょう。

●現在のECHOモードを表示します

```
ECHO ↵
```

とだけ入力します。

```
echo は<on>です
```

または、

```
echo は<off>です
```

というように、現在のECHOモードが表示されます。Human68kの初期設定では、ECHOモードはONです。

●メッセージを表示します

ECHO 〈メッセージ〉 ↵

のように、〈メッセージ〉として何か文字列を指定して実行すると、指定した〈メッセージ〉がそのまま表示されます。

バッチファイル中にこの行を入れておくと、ECHOモードがOFFであっても、バッチ処理の実行時にこの〈メッセージ〉が表示されます。

FOR

機　能

コマンドの反復実行

書　式

FOR　%%〈C〉　IN　(〈項目のセット〉)　DO　〈コマンド〉
……(バッチファイル用)
FOR　%〈C〉　IN　(〈項目のセット〉)　DO　〈コマンド〉
……(コマンド行用)

解　説

〈項目のセット〉の個数だけ、〈コマンド〉を繰り返して実行します。FOR コマンドを実行すると、変数〈C〉に〈項目のセット〉の要素が順次代入され、それに従って〈コマンド〉が実行されます。

〈C〉には1文字の変数名を指定します。ただし、バッチファイルで用いるパラメータと区別するため、数字を用いないようにしてください。

バッチファイル中で用いるときには、この変数〈C〉の前に%が2個必要(%% F のように)ですが、コマンドモードから直接入力するときには%は1個だけつけてください。

〈項目のセット〉には、次のように複数個の文字列を空白で区切って並べたものを指定します。

〈文字列〉　〈文字列〉　……

〈項目のセット〉にはワイルドカードも使えます。

〈コマンド〉には、繰り返し実行したいコマンドを指定します。このとき、このコマンドの中に FOR の直後に指定したのと同一の変数〈C〉を含ませておくことにより、〈項目のセット〉中の要素が順次これに代入され、反復実行されます。

例

```
FOR  %%E  IN  (PROC.100  PROC.200)  DO  DUMP  %%E  ↵
```

……はじめに PROC. 100、次に PROC. 200がそれぞれ%% E にセットされ、順に DUMP されます。

GOTO

内部コマンド

機　能　バッチ処理の流れの変更

書　式　GOTO　<ラベル>

解　説

GOTOコマンドは、バッチ処理を<ラベル>で定義された位置へ移します。

ラベルは8文字以内の文字列(英大文字、英小文字、数字、カタカナ、漢字など)で、必ずコロン(：)で始まっていなくてはなりません。ラベルは、バッチファイル中の任意の行に置くことができます。

GOTOコマンドで指定する<ラベル>には、先頭のコロン(：)を省いたものを指定します。<ラベル>に対応する実際のラベルがバッチファイル中にないと、バッチファイルの実行は終了します。

例

```
:LOOP
ECHO　エンドレスループ
GOTO　LOOP
```

……"エンドレスループ"を表示し続けます。実行を中止するには、CTRL＋Cを押してください。

```
:LOOP
ECHO　エンドレスループ
GOTO　LOOP 9
```

……指定したラベルがないので、バッチファイルの実行は終了します。

IF

機能　　条件設定によるバッチ処理

書式　　IF 〔NOT〕 〈条件〉 〈コマンド〉

解説

IF コマンドは、〈条件〉パラメータに従って、〈コマンド〉で指定した処理を行います。〈条件〉には、次のうちのいずれかを指定できます。〈条件〉の前に NOT を付けると、逆に指定の条件にあてはまらないときにコマンドが実行されます。

● IF 〈文字列1〉 == 〈文字列2〉 〈コマンド〉

……〈文字列1〉と〈文字列2〉が等しい場合、〈コマンド〉が実行されます。"=="の前後には必ずスペースをそれぞれ入れるようにしてください。

● IF 〈数値〉 〈コマンド〉

……〈数値〉が0でない場合、〈コマンド〉が実行されます。

● IF ERRORLEVEL 〈数値〉 〈コマンド〉

……COMMAND. X のエラーコードが〈数値〉以上の場合、〈コマンド〉が実行されます。COMMAND. X のエラーコードについては、第5章の COMMAND コマンドの項を参照してください。

● IF EXITCODE 〈数値〉 〈コマンド〉

……COMMAND. X のエラーコードが〈数値〉と等しい場合、〈コマンド〉が実行されます。COMMAND. X のエラーコードについては、第5章の COMMAND コマンドの項を参照してください。

● IF EXIST 〈ファイル名〉 〈コマンド〉

……〈ファイル名〉で指定されたファイルが存在している場合、〈コマンド〉が実行されます。

PAUSE

内部コマンド

機　能

バッチ処理の一時停止

書　式

PAUSE　〔〈コメント〉〕

解　説

PAUSE コマンドは、バッチファイルの実行を一時停止します。フロッピーディスクを交換したり、バッチ処理を中断させたりする必要のある場合に使用します。

PAUSE コマンドが実行されると、

準備ができたら何かキーを押してください

というメッセージが表示され、処理が一時停止します。再開するときは任意のキー（[CTRL]+[C]以外）を押します。

このとき[CTRL]+[C]を押すと、

バッチ処理を中止しますか？〈Y／N〉

とたずねてきます。[Y]を押すとバッチファイルの処理から抜け出し、コマンドモードに戻ります。[N]を押すと、バッチファイル中の次の行から実行が再開されます。

なお、[CTRL]+[C]は、PAUSE コマンドで一時停止しているとき以外にも有効で、バッチ処理の中止をすることができます。その場合も、同じメッセージが表示されますので、[Y]または[N]で答えてください。

〈コメント〉には、一時停止時に表示したいメッセージを指定します。たとえばここに「フロッピーディスクを交換してください」と入れておけば、

フロッピーディスクを交換してください

のようにコメントのみ表示して、バッチ処理の実行を一時停止します。

REM

機能　バッチ処理中におけるコメントの表示

書式　REM 〔<コメント>〕

解説

REMコマンドは、バッチファイルの実行中にコメントを表示したいときに使います。

REMコマンドは、<コメント>を表示するだけで他は何もしません。コメント内の区切記号は、スペース、タブ、カンマのみ使用できます。

次の例は、REMコマンドを使ったバッチファイルの一例です。

```
REM  フロッピーディスクのフォーマットおよびフロッピーディスクのコピー
REM  新しいフロッピーディスクをドライブBに挿入してください
FORMAT  B:
DIR  B:
DISKCOPY  A:  B:
```

SHIFT

内部コマンド

機　能　バッチ処理中におけるパラメータのシフト

書　式　SHIFT

解　説

SHIFT コマンドは、バッチ処理の際、仮パラメータに代入される実パラメータを、ひとつだけずらす(シフトする)ために使います。

たとえば、TEST. BAT というバッチファイルの内容が次のようになっているとします。

```
COPY  %1  %2
SHIFT
COPY  %1  %2
```

このバッチファイルを次のように実行すると、

```
TEST  X 1  X 2  X 3 ↵
```

1)　第1行の%1、%2には、それぞれ〝X 1〟、〝X 2〟が代入され、

```
COPY  X 1  X 2
```

として実行されます。

2)　第2行の SHIFT コマンドで実パラメータがひとつずれ、第3行は、

```
COPY  X 2  X 3
```

として実行されます。

このように、仮パラメータ(ここでは%1と%2)の数より実パラメータの数が多くても、SHIFT コマンドを用いてシフトすることにより、順に代入することができます。

したがって、ふつうバッチファイルで使用できるパラメータは%0から%9までの10個までですが、SHIFT コマンドを使うことで、10個以上扱えるようになります。

なお、SHIFT コマンドでシフトを行ったときに、代入すべき実パラメータがなくなると、順に空(ヌル)文字列が仮パラメータに代入されていきます。

第6章　キー入力の編集と キーボードコントロール

6．1　イントロダクション

Human68kでは、キー入力操作の能率をよくするために、テンプレート機能、ヒストリ機能（HISコマンド）、ヒストリデバイスドライバ（HISTORY.X）の3つを持っています。

●テンプレート機能

直前に実行したコマンド行を記憶しておき、その行を加工して再びキー入力として使用します。

●ヒストリ機能（HISコマンド）

それまでに入力されたコマンド行を記憶しておき、数回前に実行したコマンド行を、再び実行する際に、そのまま呼び出して使用します。

●ヒストリデバイスドライバ（HISTORY.X）

HISTORY.XはHISコマンドのヒストリ機能を大幅に拡張し、さらにエイリアス機能や簡易バッチ機能などを備えています。これを利用することでHuman68kの操作性は格段に向上します。

6．2　テンプレート機能

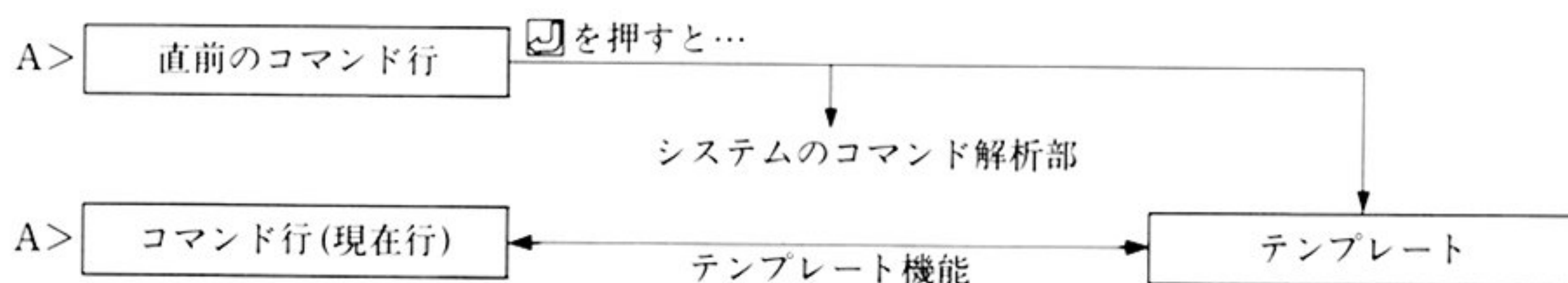
テンプレート機能は直前に入力したコマンドを〝テンプレート〟と呼ばれる場所（バッファ）に保存しておき、必要に応じて呼び出すことができるというものです。

テンプレート機能を使用すると次のような利点があります。

●同じコマンドを続けて実行するときなど、直前のキー入力を1回の操作で繰り返すことができます。

●長いファイル名などを誤って入力したとき、最初から入力し直す必要がありません。1～2回のキー操作で誤った部分を呼び出して必要なところだけ再入力できます。

つまり、直前の入力の全体または一部を活用できるわけです。

コマンド行とテンプレートの関係を図示すると、次のようになります。

ここでの動作は、次のようにして行われます。

(1)　キーボードから入力したコマンドは、リターンキーを押すまでコマンド行に保持されます。

(2) リターンキーを押すと、コマンド行がシステム内部のコマンド解析部に転送され実行されます。
同時にそのコマンド行はテンプレートにコピーされます。
このように、いったん入力したコマンド行がテンプレートにもコピーされて保持されるようになっているため、これを次回の入力時に利用できるというわけです。
次にテンプレート機能の一覧を示します。「機能名」というのは、画面のファンクションキー欄に表示されるもので、機能を省略した文字です。

テンプレート機能

キー	機能名	機　　能
F 1	C 1	テンプレートから現在行へ、1文字コピーする (COPY 1 CHARACTER)
F 2	CU	テンプレートから現在行へ、指定された文字の直前までのすべての文字をコピーする (COPY UP TO CHARACTER)
F 3	CA	テンプレートにあるすべての文字を現在行にすべてコピーする (COPY ALL CHARACTER)
F 4	S 1	テンプレート内の1文字をスキップする (コピーは行わない) (SKIP 1 CHARACTER)
F 5	SU	指定された文字の直前までテンプレート内の文字をスキップする (コピーは行わない) (SKIP UP TO CHARACTER)
F 6	VOID	現在行に入力した内容を取り消す (テンプレートの内容はそのまま)
F 7	NWL	現在行に入力した内容をテンプレートにコピーする
F 8	INS	挿入モードの設定・解除を行う
F 9	N & CU	現在行に入力した内容をテンプレートにコピーした後、指定された文字の直前までのすべての文字を現在行へコピーする
F10	EOF	CTRL + Z を入力する

次に、各キーの操作方法について、使用例をあげながら説明します。

〈F 1キー〉

"MASTERFILE.100" と入力するつもりで、次のようにファイル名を誤って入力し、エラーメッセージが表示されたとします。

TYPE　MASTERFILE.200 ↵

ここで、F 1キーを押してみます。すると、

T

と、現在行に表示されてます。このまま、続けてF 1キーを押して行くと、

TYPE　MA……

のように、順に、さきほど誤って入力した行がそのまま表示されていきます。

TYPE　MASTERFILE.

のところまで表示されたら、次に、"1" と入力し、またF 1キーを続けて押します。

TYPE　MASTERFILE.100

これで、誤りを修正することができました。ここでリターンキーを押すと、目的のファイルの内容を表示させることができます。

このようにF 1キーを使うと、いちいちタイプし直す必要がなく便利ですが、この例のように、修正したい箇所が行の後ろにあると、何度もF 1キーを押さなくてはなりません。このようなときには、次のF 2キーを使うと便利です。

〈F 2キー〉

F 2キーは、次のように使います。

TYPE　MASTERFILE.200 ↵

と、誤って入力してエラーメッセージが出たら、F 2キーを押して（この時点では何も起こりません)、続けて "2" を押します。すると、現在行に、

TYPE　MASTERFILE.

と表示されます。F 2キーにより、"2" の直前までの文字が、テンプレートから現在行にコピーされたのです。ここから後は、いままでどおり、"1"、F 1キー、F 1キーと続けて押せば、

TYPE　MASTERFILE.100

となり、修正が完了します。この方法だと、F 1キーだけを使うよりも、はるかに簡単に手直しすることができます。

〈F 3 キー〉

同じ例で、F 3 キーを使って修正してみます。

```
TYPE　MASTERFILE.200 ↵
```

と、誤って入力してエラーメッセージが出たら、F 3 キーを押します。すると現在行に、さきほどまちがえて入力した行と同じものが次のように、

```
TYPE　MASTERFILE.200
```

と表示されます。ここで、BS キー、または ← キーを押して、カーソル〝2″のところまで移動して〝200″を消し、〝1″を入力します。そして、このまま再び F 3 キーを押すと、現在行は、

```
TYPE　MASTERFILE.100
```

と表示され、修正が完了します。

〈F 4 キー〉

F 4 キーは、次のようなときに使います。〝FILE3.TXT″をコピーしようとして、

```
COPY　FILE34.TXT　B: ↵
```

と入力してしまい、エラーが出力されてしまったとします。このとき、まず F 1 キー、F 2 キーなどを使って、次の現在行に、〝3″のところまで表示させます。

```
COPY　FILE3
```

ここで、F 4 キーを押すと、テンプレートの中の内容が1文字（ここでは〝4″）だけスキップ（飛び越し）されます。このとき、画面上では何もおこりませんが、次に残りの文字を F 3 キーを押して現在行にコピーすると、

```
COPY　FILE3.TXT　B:
```

となります。これで、修正が完了します。

〈F 5 キー〉

F 5 キーは、テンプレート中の指定の文字までをスキップするもので、使い方は F 2 キーと似ています。

```
COPY　FILE001.DOC　B:
```

という行がテンプレート中にあるとして、F 1 キー、F 2 キーなどを使い、次の位置まで現在行にコピーしたとします。

COPY　FILE

ここで、F 5 キー、続いて〝．〟を押すと、テンプレート中の〝001〟がスキップされます。F 3 キーを押せば、

COPY　FILE.DOC　B：

となります。

〈F 6 キー〉

F 6 キーは、いま入力している現在行の内容をすべて取り消すのに使います。テンプレートの内容は以前のコマンド行で入力したままとなっています。テンプレート機能による作業中に、コマンド行の編集・修正が失敗したときなど、元のイメージにもどすときに使用すると便利です。

〈F 7 キー〉

F 7 キーは、以前のテンプレートの内容を捨て、現在行に入力されている内容をそのままテンプレートに移します。このとき、カーソルは次行の最初に戻り、何も入力されていない状態となります。何かコマンドを入力し、リターンキーを押す前に誤入力に気がついたときは、このキーを使うとエラーになる前に修正を行うことができます。

〈F 8 キー〉

F 8 キーは、テンプレート中の内容の、任意の場所に文字を追加挿入したいときに使います。

TYPE　FLE.TXT

というコマンドテンプレートに入っているとして、F 1 キーなどを使って次の位置まで現在行にコピーしたとします。

TYPE　F

ここで、F 8 キー、続いて〝I〟を押すと、〝F〟の後に〝I〟が挿入されます。F 3 キーを押せば、

TYPE　FILE.TXT

となります。

なお、F 8 キーは、いちど押すと挿入モードになりますが、もういちど押すか、または別のテンプレート機能キーを押すことで挿入モードを解除することができます。

〈F 9 キー〉

F 9 キーは、F 2 キーと同様の機能を、テンプレートに対してではなく、現在行に対して行います。現在行で、次のように入力した時点で、

TYPE　FILD.TXT

[F 9]キー、続いて“D”キーを押せば、現在行に、

TYPE　FIL

と表示されます。このとき、テンプレートには、さきに入力した現在行の全文字がコピーされています。

このキーは、[F 7]キーと同様、リターンキーを押す前に誤りを修正したいときに便利です。

6.3　ヒストリ機能 (HIS コマンド)

ヒストリ行列とは、それまでに実行したコマンド行を、システム内のバッファに保存したものです。

ヒストリ機能は、このヒストリ行列を利用することにより、それまでに実行したコマンドをさかのぼって呼び出すことのできる機能です。

まず、実際にヒストリ行列を表示してみましょう。次のように入力します。

HIS [↵]

これで、たとえば次のように、それまでに実行したコマンドが順に表示されます。

```
00005 : DIR    B :
00004 : TYPE    B : FILE.100
00003 : COPY    B : FILE.100    A : FILE.OK
00002 : DIR
00001 : DEL    B : FILE.100
00000 : HIS
```

左側には、いま入力した“HIS”コマンドから逆順に数えて“いくつ前の”コマンド行であるかが数字で示され、“：”以下右側に保存されているコマンド行が表示されます。HIS コマンドについて、くわしくは「第5章　コマンド」の HIS の項をご覧ください。

このヒストリ行列を実際に利用してヒストリ機能を使うには、次に示すようなキー操作を行います。

－〈数値〉 [↵]　　注目行から逆方向(古い方向)へ〈数値〉だけ移動し、その内容をテンプレートに格納します。〈数値〉を省略すると、1を指定したものとみなされます。

＋〈数値〉 [↵]　　注目行から正方向(新しい方向)へ〈数値〉だけ移動し、その内容をテンプレートに格納します。〈数値〉 を省略すると、1を指定したものとみなされます。

－／〈文字列〉 [↵]　　注目行から逆方向にある〈文字列〉で始まる行に移動し、その内容をテンプレートに格納します。

＋／〈文字列〉 [↵]　　注目行から正方向にある〈文字列〉で始まる行に移動し、その内容をテンプレートに格納します。

－？〈文字列〉 [↵]　　注目行から逆方向にある〈文字列〉を含む行に移動し、その内容をテンプレートに格納します。

＋？〈文字列〉 ⏎	注目行から正方向にある〈文字列〉を含む行に移動し、その内容をテンプレートに格納します。
⏎	最新の行に移動し、その内容をテンプレートに格納します。
ROLL DOWN	注目行から逆方向へ1つだけ移動し、その内容をテンプレートに格納します。
ROLL UP	注目行から正方向へ1つだけ移動し、その内容をテンプレートに格納します。
UNDO	現在のテンプレート中の内容を、そのままコマンド行として実行します。F 3 ⏎を押したのと効果は同じです。

ここで、注目行とは、ヒストリ行列中で、いまシステムが注目している行のことを指します。

前記の例をもとに、実際にヒストリ機能を用いて作業を行ってみます。

ROLL DOWNを1回押すか、または、

　－ ⏎

と入力します。すると、次のコマンド行に、

```
A>DEL  B:FILE.100
```

と表示されます。注目行がヒストリ行列中を、逆方向に1つだけ移動したわけです。ここでもういちど同じ操作を行うと、さらに注目行が1つ逆に移動し、次のコマンド行に、

```
A>DIR
```

と表示されます。

次に、ROLL UPを押すか、または、

　＋ ⏎

と入力すると、次のコマンド行に、

```
A>DEL  B:FILE.100
```

と表示されます。こんどは、前の操作とは反対に、注目行がヒストリ行列中を正方向に移動しました。次に、

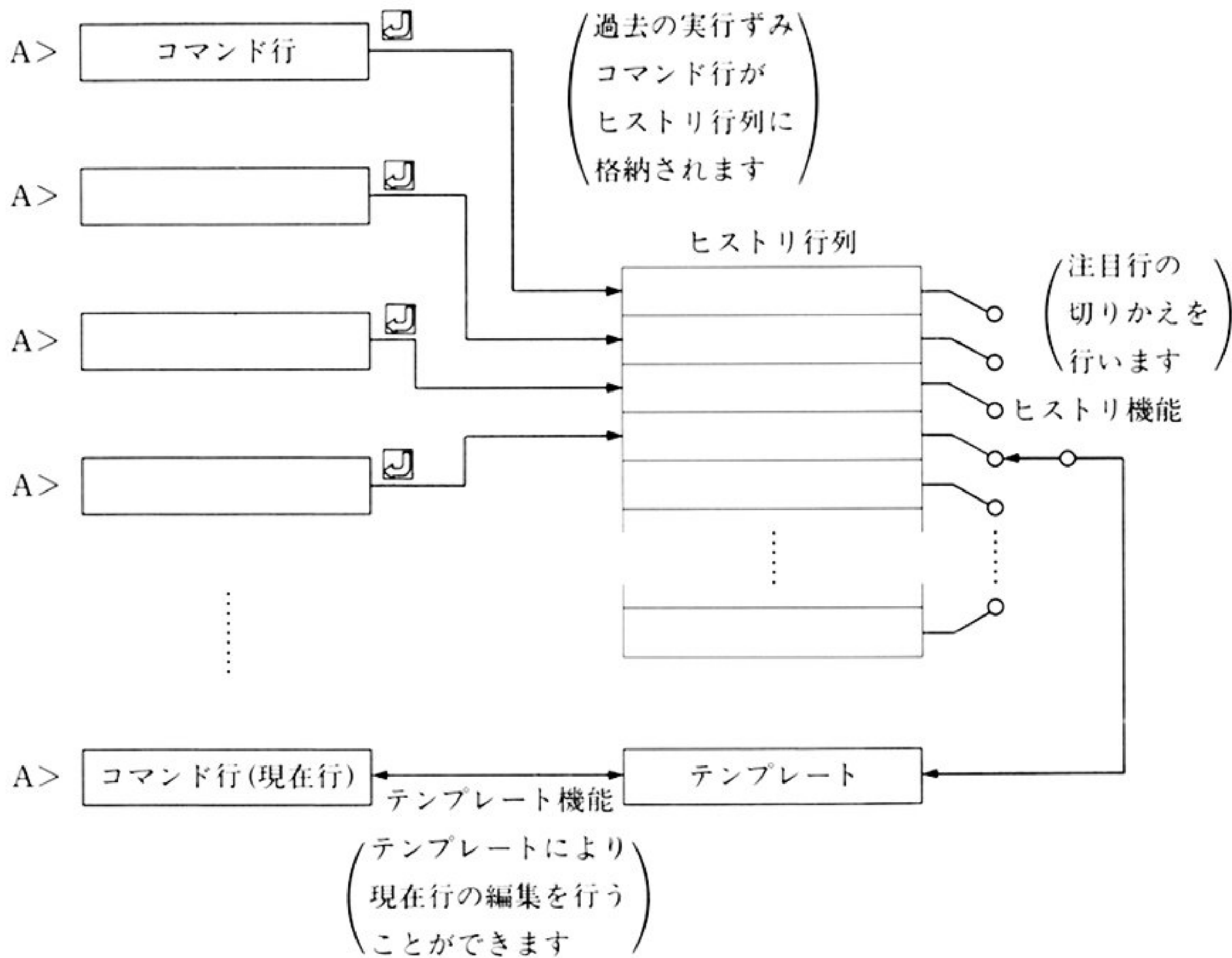
　－／TYPE ⏎

と入力してみると、ヒストリ行列中で、"TYPE"で始まる行が逆方向にさがされ、

```
A>TYPE  B:FILE.100
```

と表示されます。

以上どの場合も、ヒストリ行列の注目行は、表示されると同時にその内容がテンプレート中に入り、カーソルはさらにその次の行に移ります。

さきに注目行を、「システムが注目している行」と説明しましたが、実際には、ヒストリ行列中で、いまテンプレートに格納されているヒストリ行、およびその行番号を指すことになります。

したがって、この状態でテンプレート機能を使うことにより、いま表示された行をコマンド行として利用できるのです。

たとえば、ROLL UP を押して、次の行を表示させたとします。

```
A>COPY　B:FILE.100　A:FILE.OK
A>■
```

ここでF 2 を押し、続けて"A"を押せば、テンプレートに入れられたこのヒストリ行が編集されて、次のように表示されます。

```
A>COPY　B:FILE.100　A:FILE.OK
A>COPY　B:FILE.100 ■
```

ここで、他のテンプレート機能用のキーも利用すれば、このコマンド行を自由に編集することができるわけです。

こうして、ヒストリ機能とテンプレート機能をあわせて利用することにより、キー入力を効率よく行うことができます。

なお、ヒストリ行列はバッファに保存されますが、バッファの領域の限度を超えた部分については保存することはできません。ヒストリ行列の格納量を増やしたいときはバッファの領域を増やす必要があります。このときは、CONFIG.SYSファイル中に指定されたSHELL行中のCOMMAND.Xに、"/H"スイッチをつけてヒストリエリアサイズを指定してください。詳細は、第5章の「COMMAND」コマンドの項を参照してください。

6.4 ヒストリデバイスドライバ

Human68kの操作環境を格段に向上させるために、"HISTORY.X"というデバイスドライバを用意しています。ここでは、このHISTORY.Xの使い方について解説します。6.3で解説したヒストリ機能はCOMMAND.Xの内部コマンドのHISコマンドで行いましたが、ここで解説するヒストリデバイスドライバはCONFIG.SYSで組み込むものです（コマンド行からも可能)。実際の組み込み方法は後ほど解説します。

6.4.1 HISTORY.Xの概要

このデバイスドライバは、キーボードのタイプ量を極力減らすために次のような機能を持っています。

1. コマンド行での編集機能
2. テンプレートの操作
3. 入力文字列のヒストリの検索と再入力（ヒストリ機能)
4. ディレクトリ変更履歴の検索と再入力
5. エイリアス（別名定義）機能
6. 簡易バッチ機能
7. 環境変数の取り込み・その他

各機能の概要は次のとおりです。

1. コマンド行での編集機能

コマンドやパラメータを間違えて入力した場合、エディタと同じように入力した行を編集することができます。たとえば、

```
TYPE ¥HELP¥HST.HLP■
```

と入力して、HSTはHISTの間違いだと気付いたときは、カーソル移動キーでカーソルを"S"の位置に移動して"I"を入力すれば修正できます。このカーソル移動の方法は、基本的にスクリーンエディタ"ED"と同じです。

2. テンプレートの操作

テンプレート機能はHISTORY.Xが組み込まれていなくても利用できますが、HISTORY.Xではさらにテンプレートの引き数だけをコピーする機能が追加されます。

3. 入力文字列のヒストリの検索と再入力（ヒストリ機能)

HISコマンドでも、これまで入力したコマンドをさかのぼって呼び出すことができます。しかし、HISコマンドの場合、テンプレートに格納したり、あるいはバッチファイルにしたり、といった複数の手続きが必要となります。ところが、HISTORY.Xでは、以前に入力したコマンドや引き数をワンタッチで呼び出して実行することができます。

これまでに

```
ED ¥HELP¥HHIST.BAT
TYPE ¥HELP¥HHIST.BAT
HHIST
```

というコマンドを入力していた場合、もう一度HHIST.BATを表示したいときは、 CTRL キーを押しながら W を押します。こうすると、コマンド行には

```
A>HHIST
```

と表示されます。ここで、さらに CTRL キーを押しながら W を押すと

```
TYPE ¥HELP¥HHIST.BAT
```

と表示されるので、ここでリターンキーを押せばTYPEコマンドが実行されます。

4．ディレクトリ変更履歴の検索と再入力

ディレクトリの移動はCDコマンドで行いますから、ヒストリ機能を利用しても以前に入力したコマンドを呼び出すことができます。しかし、ハードディスクを利用している場合などは、ディレクトリを移動することが多いので、ヒストリ機能とは別にこの機能を用意しています。

これまでに

```
CD ¥ETC¥GAME
CD ¥PRG¥CSRC¥GREP
CD ¥UTL¥TOOL
```

とディレクトリを変更していた場合、¥ETC¥GAMEにディレクトリを移動するときは、 CTRL キーを押しながら Q を2回押します。こうすると、コマンド行に

```
A>CD ¥ETC¥GAME
```

と表示されるので、ここでリターンキーを押せばCDコマンドが実行されて目的のディレクトリに移動できます。

5．エイリアス（別名定義）機能

コマンドを別の名前で定義する機能です。スイッチやオプションの多いコマンドはなかなかおぼえにくいものですが、こういったコマンドを短い名前で定義しておけば、いちいちマニュアルを見なくても簡単に実行できるようになります。たとえば、

```
SPEED 1200 B8 PN S1 XON
```

を〝RSINIT〟で定義しておけば、

```
RSINIT ↵
```

と入力するだけで、

```
SPEED 1200 B8 PN S1 XON
```

が実行されます。また、マルチ処理やリダイレクトと組み合わせたり、バッチ処理と同じようにパラメータも利用することができます。

6．簡易バッチ機能

入力したコマンドはヒストリバッファに格納されますが、これをバッチ処理のように連続して実行できます。たとえば、ヒストリバッファに次のように格納されているときに

```
{1} FORMAT B:
{2} DISKCOPY A: B:
{3} CHKDSK B:
{4} ED ¥HELP¥HHIST.BAT
{5} TYPE ¥HELP¥HHIST.BAT
{6} HHIST
{7} CHKDSK A:
```

A>1－3　7■

↓ [CTRL] ＋ [N]　（[1] [－] [3] [スペース] [7] [CTRL] ＋ [N] と入力する）

と入力すると、1から3番目のコマンドと7番目のコマンドが順次実行されます。

7．環境変数の取り込み・その他

コマンド行に環境変数を展開したり、HISTORY.Xの状態を切り替えるための機能です。

6．4．2　HISTORY.Xの機能

ここでは、HISTORY.Xの全ての機能を、例をまじえながら解説していきます。HISTORY.Xのコマンドには、CTRL（コントロールキー）系とESC（エスケープキー）系の2種類あります。

◆CTRL系のコマンド

CTRL系のコマンドは、[CTRL] キーを押しながら1文字のキーを押すことによって入力するものです。この操作方法を次のようにプラス記号でつないで表記します。

[CTRL] ＋ [X]（Xは1文字のキー）

◆ESC系のコマンド

ESC系のコマンドは、[ESC] を押し、続けて1文字のキーを押すことによって入力するものです。この操作方法を次のように〝・〞記号で区切って表記します。

[ESC]・[X]（Xは1文字のキー）

◆表記について

コマンドの解説で使用する記号は、それぞれ次のような意味です。

記　号	意　　味
' '	スペース（空白文字）
〈str〉	任意の文字列
〈num〉	数値
〈item〉	8文字以下の文字列
〈env〉	任意の環境変数名
..	繰り返しを示す

1．コマンド行での編集機能

編集機能はカーソルの移動と文字の挿入／削除の2つに分けて説明します。表中の「キー」のところに複数のキーが書かれているものは、どちらのキーを使用しても同じです。

・カーソルの移動

キー	機　能
CTRL + S 、 ←	カーソルを1文字左へ移動
CTRL + D 、 →	カーソルを1文字右へ移動
CTRL + A 、 ↑	カーソルを1ワード左へ移動
CTRL + F 、 ↓	カーソルを1ワード右へ移動
CTRL + B	カーソルを行頭／行末へ移動

表中の「ワード」というのは、次のような文字で区切られた文字列のことです。

タブ　スペース　$　%　¥　(　)　,　.　／　:　;　<　=　>　[　]　_　{　}　|
「　」　、　。　・

例)

```
A>CD  ¥DOC¥REPORT¥X68K■
              ↓ CTRL + A
A>CD  ¥DOC¥REPORT¥X68K
              ↓ CTRL + B
A>CD  ¥DOC¥REPORT¥X68K
              ↓ CTRL + B
A>CD  ¥DOC¥REPORT¥X68K■
```

・文字の挿入／削除

キー	機能
CTRL + H 、 BS	カーソル位置の左の文字を削除
CTRL + G 、 DEL	カーソル位置の文字を削除
CTRL + ^	カーソルの左1ワードを削除
CTRL + ¥	カーソルの右1ワードを削除
CTRL + U	カーソルより左の文字列を削除
CTRL + K	カーソル位置より行末までを削除
CTRL + O	挿入／上書きモードの切り替え
CTRL + [、 ESC	コントロール文字の入力
TAB	入力文字のファイル名保管

文字の入力時は、基本的に挿入モードになっています。途中で挿入／上書きのモードを変えても、次に入力するときは挿入モードになります。ただし、この状態は起動時のオプションやコマンドで切り替えることもできます（詳しくは後述します)。

コントロール文字を入力するときは CTRL ＋ [（または ESC ）を押した後に、CTRL キーとコントロール文字を押します。

例1）

```
A>ECHO ■
        ↓ CTRL + [
A>ECHO ■
        ↓ CTRL + L  (コントロール文字)
A>ECHO ^L■
```

例2）

```
A>CD ¥DOC¥REPORT¥X68K■
                ↓ CTRL + A
A>CD ¥DOC¥REPORT¥X68K
                ↓ CTRL + ^
A>CD ¥DOC¥X68K
```

例3）

```
A>CD  ¥DOC¥RE■
            ↓ [TAB]
A>CD  ¥DOC¥REPORT■
```

2．テンプレートの操作

テンプレートの操作には、ESC系のコマンドを使います。

・テンプレート機能

キー	機能
ESC・S (C1)	テンプレートから1文字コピー
ESC・T (CU)	テンプレートから指定文字までコピー
ESC・U (CA)	テンプレートから残りの文字をすべてコピー
ESC・V (S1)	テンプレート上で1文字スキップ
ESC・W (SU)	テンプレート上で指定文字までスキップ
ESC・E (VOID)	入力行をすべて削除
ESC・J (NWL)	入力行をテンプレートにコピー
ESC・P (INS)	テンプレートの挿入／上書きモードの切り替え
ESC・F (NWL&CU)	入力行をテンプレートにコピーした後、指定文字までコピー
ESC・0 〜 ESC・9	テンプレートの1〜10番目の引数をコピー

表中のカッコ書きのものは、「6.2　テンプレート機能」で示した機能名のことです。ファンクションキーによるテンプレート機能もそのまま機能しますので、どちらか使いやすい方を使ってください。

「引数」というのは、タブやスペースで区切られた文字列を示します。

例）

テンプレートに

```
TYPE  ¥DOC¥REPORT¥ABC.DOC
```

という文字列が入っているものとすると、次のような操作ができます。

```
A>DEL ■
       ↓ [ESC]・[1]
A>DEL  ¥DOC¥REPORT¥ABC.DOC■
```

3．入力文字列のヒストリの検索と再入力（ヒストリ機能）

ヒストリ機能は、以前に入力したコマンドを再度入力するための機能です。新たに入力したコマンドは、特別な指定をしないかぎり、自動的にヒストリバッファと呼ばれる領域に登録されます。

・ヒストリ機能

キー	機能
CTRL + W 、ROLLDOWN	ヒストリの新しいものから検索・再入力
〈str〉 CTRL + W 、 〈str〉 ROLLDOWN	〈str〉で始まるヒストリを新しい方から検索・再入力
' '〈str〉 CTRL + W 、 ' '〈str〉 ROLLDOWN	〈str〉を含むヒストリを新しい方から検索・再入力
CTRL + E 、ROLLUP	ヒストリの古いものから検索・再入力
〈str〉 CTRL + E 、 〈str〉 ROLLUP	〈str〉で始まるヒストリを古い方から検索・再入力
' '〈str〉 CTRL + E 、 ' '〈str〉 ROLLDOWN	〈str〉を含むヒストリを古い方から検索・再入力
CTRL + R	ヒストリの新しいものから引数を検索・再入力
〈str〉 CTRL + R	ヒストリの新しいものから〈str〉を含む引数を検索・再入力
〈str〉' ' CTRL + R	ヒストリの新しいものから引数を検索し、〈str〉' 'の後に再入力
CTRL + T	ヒストリの古いものから引数を検索・再入力
〈str〉 CTRL + T	ヒストリの古いものから〈str〉を含む引数を検索・再入力
〈str〉' ' CTRL + T	ヒストリの古いものから引数を検索し、〈str〉' 'の後に再入力
ESC ・ ^	ヒストリの新しいものから引数部分を検索・再入力
ESC ・ ¥	ヒストリの古いものから引数部分を検索・再入力
CTRL + L	ヒストリの新しいものを１画面分表示
〈str〉 CTRL + L	ヒストリの新しいものから〈str〉を含むものをすべて表示
ESC ・ L	ヒストリをすべて表示
〈str〉 ESC ・ L	〈str〉を含むヒストリをすべて表示
〈num〉 ESC ・ N	〈num〉番目のヒストリの再入力
ESC ・ @	入力中の行をヒストリに登録し、入力行を削除

（注）引数部分とは、コマンド行に入力された文字列のうち、コマンドを除いたスペース以降の部分です。

例えば

```
COPY  A : TEST1.DAT  B :
```

では、

```
A : TEST1.DAT  B :
```

という部分を引数部分といいます。

ヒストリバッファに次のような文字列が入っているものとして操作例を示します。

```
{1}  ED ¥HELP¥HHIST.BAT
{2}  TYPE  ¥HELP¥HHIST.BAT
{3}  TYPE  REPORT.DOC
{4}  HHIST
{5}  CD  ¥DOC¥REPORT¥X68K
```

例1）

```
A>■
        ↓ [CTRL] + [W]
A>CD  ¥DOC¥REPORT¥X68K■
        ↓ [CTRL] + [W]
A>HHIST■
```

例2）

```
A>TY■
        ↓ [CTRL] + [W]
A>TYPE  REPORT.DOC■
        ↓ [CTRL] + [W]
A>TYPE  ¥HELP¥HHIST.BAT■
```

例3）

```
A>  HE■
        ↓ [CTRL] + [W]
A>TYPE  ¥HELP¥HHIST.BAT■
        ↓ [CTRL] + [W]
A>ED ¥HELP¥HHIST.BAT■
```

例4）

```
A>  DOC■
        ↓ CTRL + E
A>TYPE  REPORT.DOC
        ↓ CTRL + E
A>CD  ¥DOC¥REPORT¥X68K
```

例5）

```
A>■
        ↓ CTRL + R
A>¥DOC¥REPORT¥X68K■
        ↓ CTRL + R
A>CD■
```

例6）

```
A>DO■
        ↓ CTRL + R
A>¥DOC¥REPORT¥X68K■
        ↓ CTRL + R
A>REPORT.DOC■
```

例7）

```
A>DIR  ■
        ↓ CTRL + R
A>DIR  ¥DOC¥REPORT¥X68K■
```

例8）

```
A>DEL ■
        ↓ ESC ・ ^
A>DEL  ¥DOC¥REPORT¥X68K■
        ↓ ESC ・ ^
A>DEL  REPORT.DOC■
```

例9）

```
A>2■
        ↓ ESC ・ N
A>TYPE  ¥HELP¥HHIST.BAT■
```

例10)

```
A>ED  SAMPLE.DOC■
        ↓ [ESC]・[@]
A>■        ("ED  SAMPLE.DOC" はヒストリに登録される)
        ↓ [CTRL]+[L]

|1|  ED ¥HELP¥HHIST.BAT
|2|  TYPE  ¥HELP¥HHIST.BAT
|3|  TYPE  REPORT.DOC
|4|  HHIST
|5|  CD  ¥DOC¥REPORT¥X68K
|6|  ED  SAMPLE.DOC

>■
```

4．ディレクトリ変更履歴の検索と再入力

以前にCDコマンドで変更したディレクトリのパス名を再度入力する機能です。CDコマンドでディレクトリの変更を行うと、その内容は自動的にヒストリバッファとディレクトリ変更履歴の両方に登録されます。

・ディレクトリ変更履歴の検索

キー	機能
[CTRL]+[Q]	ディレクトリの変更履歴を新しいものから検索・再入力
<str> [CTRL]+[Q]	<str>を含むディレクトリの変更履歴を新しいものから検索・再入力
<str>' '[CTRL]+[Q]	ディレクトリの変更履歴を新しいものから検索し、<str>' 'の後にディレクトリ部を再入力
[CTRL]+[Y]	ディレクトリの変更履歴を古いものから検索・再入力
<str> [CTRL]+[Y]	<str> ディレクトリの変更履歴を古いものから検索・再入力
<str>' '[CTRL]+[Y]	ディレクトリの変更履歴を古いものから検索し、<str>' 'の後にディレクトリ部を再入力
[CTRL]+[_]	ディレクトリの変更履歴をすべて表示
<str> [CTRL]+[_]	<str>を含むディレクトリの変更履歴をすべて表示

ディレクトリの変更履歴に次のような文字列が入っているものとして操作例を示します。

```
CD　A：¥DOC¥REPORT
CD　B：¥DOC¥X68K
CD　A：¥ETC¥GAME
CD　B：¥TOOL¥TOOL1
```

例1）

```
A>■
      ↓ [CTRL]+[Q]
A>CD　B：¥TOOL¥TOOL1■
      ↓ [CTRL]+[Q]
A>CD A：¥ETC¥GAME■
```

例2）

```
A>DO■
      ↓ [CTRL]+[Q]
A>CD　B：¥DOC¥X68K
      ↓ [CTRL]+[Q]
A>CD　A：¥DOC¥REPORT
```

例3）

```
A>DIR　■
      ↓ [CTRL]+[Q]
A>DIR　B：¥TOOL¥TOOL1■
      ↓ [CTRL]+[Q]
A>DIR　A：¥ETC¥GAME■
```

例4）

```
A>RD　DO■
      ↓ [CTRL]+[Q]
A>RD　B：¥DOC¥X68K
      ↓ [CTRL]+[Q]
A>RD　A：¥DOC¥REPORT
```

5．エイリアス（別名定義）機能

コマンドを別の名前で定義する機能です。定義した名前はエイリアス（別名）バッファに登録されます。

・エイリアス機能

キ　ー	機　　能
<item>' '<str> ESC ・ I	<str> を <item> で別名定義。%変数も使用可能
<item> ESC ・ I	<item> の別名定義を削除
－<item>' '<item>‥ ESC ・ I	複数の <item> の別名定義を無効にする
ESC ・ I	別名定義の一覧を表示

エイリアス機能は、バッチファイルと同様に%1～%9の変数や環境変数を利用することができます。

例1)

A>LS　DIR　%1　／N　／P■
　　　　　　　　ESC ・ I （Iはアルファベット）
　　　　　↓　（これで "DIR　%1　／N　／P" が "LS" として定義されます）

A>■

ここで

A>LS　￥DOC　↵

と入力すると、

DIR　￥DOC　／N　／P

が実行されます。また、マルチ処理機能、リダイレクト、パイプを利用してさらに複雑なコマンドを定義することができます。

例2)

A>VF　VOL　%1　||　TREE　／F　%1￥　|　FIND　%2■
　　　　　↓ ESC ・ I

この例では、ドライブ名と検索する文字をパラメータで与えて実行します。
たとえば、

A>VF　B：　"DOC"　↵

と入力すると、

VOL　B：　||　TREE　／F　B：￥　|　FIND　"DOC"

が実行されます。

6．簡易バッチ機能

ヒストリに登録されているコマンドを、バッチ処理のように連続して実行する機能です。一度バッチ機能を使うと、その内容はバッチバッファと呼ばれる領域に格納され、次からはさらに簡単に実行できるようになります。

・簡易バッチ機能

キー	機　能
<num> ' ' <num> ‥ CTRL ＋ N	<num> ‥で指定されたヒストリをバッチバッファに登録し、実行
CTRL ＋ N	バッチバッファに登録された簡易バッチを実行

ヒストリに次のような文字列が入っているものとして操作例を示します。

```
{1}  ED  ¥HELP¥HHIST.BAT
{2}  TYPE  ¥HELP¥HHIST.BAT
{3}  TYPE  REPORT.DOC
{4}  HHIST
{5}  CD  ¥DOC¥REPORT¥X68K
{6}  ED  SAMPLE.DOC
```

例)

```
A>2  5－6■
      ↓ CTRL ＋ N

{2}  TYPE  ¥HELP¥HHIST.BAT
{5}  CD  ¥DOC¥REPORT¥X68K
{6}  ED  SAMPLE.DOC

  実行しますか <Y／N> ■
```

ここで、 Y またはリターンキーを押すと実行されます。その他のキーを押すと実行しません。また、実行中に中断したいときは、実行中断待ち時間中に任意のキーを押します。

簡易バッチを実行すると、その内容はバッチバッファに格納されるので、次に同じ内容を実行したいときは、 CTRL ＋ N だけ押します。これで、簡易バッチの内容が表示され、同じように確認のメッセージが表示されるので、これに応えれば実行できます。

7．環境変数の取り込み・その他

コマンド行に環境変数を展開したり、HISTORY.Xの状態を切り替えるための機能です。

・環境変数の取り込み・その他

キー	機能
% <env> % ESC・／	% <env> %を環境変数 <env> の内容に置き替える
ESC・＊	現在入力中の行にかぎって、別名と環境変数の展開のON／OFF切り替え（注1)
CTRL＋］	カーソル位置の文字を大文字かまたは小文字に変換する
ESC・］	大文字・小文字変換で大文字にするか小文字にするかを切り替える
ESC・O	編集機能の挿入・上書きの状態を保存する／保存しないを切り替える
ESC・－	－と＋で始まる行をヒストリに登録しない
ESC・＋	－と＋で始まる行をヒストリに登録する
HELP	HISTORY.Xの使い方を表示する

(注1)

ESC・＊を入力すると、現在入力中の行についてのみ別名と環境変数の展開を行わないようになります。その次からは展開を行うようになります。

例)

PATHが

A：¥BIN；A：¥ETC；A：¥TOOL

のように設定されているときに、¥ETCを¥UTLに変えたいときは、次のように入力します。

A＞PATH　%PATH%■
　　↓　ESC・／
A＞PATH　A：¥BIN；A：¥ETC；A：¥TOOL■
　　↓　（ここで編集機能を利用して書き替えます）
A＞PATH　A：¥BIN；A：¥UTL；A：¥TOOL ↵

なお、単にPATHにA：¥UTLを追加するときは次のように入力します。

A＞PATH　%PATH%；A：¥UTL ↵

6.4.3 定義ファイルについて

ヒストリ、エイリアス、簡易バッチ、ディレクトリ変更履歴は、あらかじめファイルとして定義しておくことで、起動時から利用できるようになります。このファイルを「ヒストリ定義ファイル」と呼びます。この他にファンクションキーなどの設定を行う「キー定義ファイル」もあります。

1．ヒストリ定義ファイル

ヒストリ定義ファイルは"HISTORY.HIS"というファイル名で、標準的なテキストファイルですからエディタで編集・作成することができます。なお、次のような行は制御用として使用されるため、特別な意味を持ちます。

```
＃＃＃ HIS01 ＃＃＃    … 1番目のヒストリ定義の始まりを示す
＃＃＃ HIS02 ＃＃＃    … 2番目のヒストリ定義の始まりを示す
＃＃＃ BAT01 ＃＃＃    … 1番目の簡易バッチ定義の始まりを示す
＃＃＃ BAT02 ＃＃＃    … 2番目の簡易バッチ定義の始まりを示す
＃＃＃ ALIAS ＃＃＃    … エイリアス定義の始まりを示す
＃＃＃ CHDIR ＃＃＃    … ディレクトリ変更履歴定義の始まりを示す
```

ヒストリ定義ファイルには制御行がなくてもかまいませんが、その場合は先頭に"＃＃＃ HIS01 ＃＃＃"があるものとして扱われます。また、ヒストリ機能の定義の中には、エイリアスで定義されたコマンドも書くことができます。

ヒストリ定義ファイルの例を次に示します。

```
＃＃＃ HIS01 ＃＃＃
ED ￥HELP￥HHIST.BAT
TYPE ￥HELP￥HHIST.BAT
TYPE REPORT.DOC
HHIST
CD ￥DOC￥REPORT￥X68K
VF A：￥ "SYS"
＃＃＃ ALIAS ＃＃＃
LS DIR ％1 ／N ／P
VF VOL ％1 ｜｜ TREE ／F ％1￥ ｜ FIND ％2
C CC ／L ％1
＃＃＃ CHDIR ＃＃＃
CD A：￥DOC￥REPORT
CD B：￥DOC￥X68K
CD A：￥ETC￥GAME
CD B：￥TOOL￥TOOL1
```

なお、各行の最後は必ず〈改行〉（リターンキーを押すと入る）で区切ります。

2．キー定義ファイル

キー定義ファイルのファイル名は"KEY.HIS"で、KEYコマンドで作成します。KEYコマンドでは"KEY.SYS"というファイルも作成できますが、これにはHISTORY.Xを使用しないときのキー（標準的なキー定義）を設定し、"KEY.HIS"にはHISTORY.Xを使用するときのキーを設定

します。これによって、HISTORY.Xを使用する場合としない場合で、自動的にキー定義を切り替えることができます。キー定義ファイルの作り方については、KEYコマンドを参照してください。

キー定義ファイルの例を次に示します。

F01 － F10	標準（KEY.SYS）と同じ
F11 － F20	標準（KEY.SYS）と同じ
F21 (ROLL UP)	^E
F22 (ROLL DOWN)	^W
F23 (INS)	〈esc〉P
F24 (DEL)	^G
F25 (↑)	^A
F26 (←)	^S
F27 (→)	^D
F28 (↓)	^F
F29 (CLR)	CLS〈cr〉
F30 (HELP)	^V
F31 (HOME)	<esc>E
F32 (UNDO)	<esc>U 〈cr〉

ただし、キー定義ファイルはテキストファイルではありませんので、TYPEコマンドなどで内容を確認することはできません。

6.4.4　オプションについて

HISTORY.Xは、通常はデバイスドライバとして組み込みますが、コマンド行から起動することもできます。このときのオプションには、次の3種類あります。

1. 組み込み時（初期起動時）に設定できるもの
2. 随時設定変更できるもの（初期組み込み時にも設定可能）
3. 組み込み時以外にのみ設定できるもの

以下にオプションの一覧を示しますが、矛盾するオプションを指定した場合、後に指定した方が有効になります。また、各オプションは1個以上のスペースで区切って指定します。

1．組み込み時（初期起動時）に設定できるもの

オプション	内　　　容
／AR	エイリアスの中でさらにエイリアスを使用できるようにする。 デフォルトでは使用できない。
／D〈パス名〉	ヒストリ定義ファイルとキー定義ファイルの格納されている位置を示す。 デフォルトは￥HIS￥と同じ位置。
／SA〈サイズ〉	エイリアス用のバッファに使用するメモリの大きさをKB単位で指定する。 デフォルトは2KB。
／SB〈サイズ〉	簡易バッチ用のバッファに使用するメモリの大きさをKB単位で指定する。 デフォルトでは2KB。簡易バッチバッファの個数はヒストリバッファの個数と同数確保される。なお、各バッファの大きさは〈サイズ〉で指定した大きさになる。
／SC〈サイズ〉	ディレクトリ変更履歴用のバッファに使用するメモリの大きさをKB単位で指定する。 デフォルトでは2KB。
／SH〈個数〉，〈サイズ〉，〈サイズ〉，・・	ヒストリ用のバッファの個数とそれぞれに使用するメモリの大きさをKB単位で指定する。 〈個数〉のデフォルトは2で、1～64まで指定可能。 〈サイズ〉は各バッファの大きさを指定する。省略すると32KBずつ設定される。

2．随時設定変更できるもの（初期組み込み時にも設定可能）

オプション	内　　容
／*N	エイリアス、環境変数の展開を行わない。
／*Y	エイリアス、環境変数の展開を行う（デフォルト）。
／*	上記の2つを切り替える。
／－	－と＋で始まる行をヒストリに登録しない（デフォルト）。 コマンドプロセッサとしてCOMMAND.Xを使用したときに設定する。
／＋	－と＋で始まる行をヒストリに登録する。 コマンドプロセッサとしてCOMMAND.X以外を使用したときに設定する。
／＋－　または　／－＋	上記の2つを切り替える。
／BB	ヒストリ用のバッファを入力バッファの位置（アドレス）で切り替える（デフォルト）。
／BP	ヒストリ用のバッファをプロセスの位置（アドレス）で切り替える。
／B	上記の2つを切り替える。
／FY	HISTORY.Xの使用を中止／再開を切り替えたときに、ファンクションキーの定義を変更する（デフォルト）。 HISTORY.Xの使用中はキー定義ファイルとして"KEY.HIS"の内容を割り当て、使用を中止したときは"KEY.SYS"の内容を割り当てる。
／FN	HISTORY.Xの使用を中止／再開を切り替えたときに、ファンクションキーの定義を変更しないようにする。
／F	上記の2つを切り替える。
／FS	現在のファンクションキーの内容をHISTORY.Xに登録する。 HISTORY.Xの使用時／未使用時のどちらに登録するかは、現在の状態によって決まるが、／Kと／Uのオプションと組み合わせることでさまざまな指定ができる。たとえば、"／K　／FS　／U"の順で指定すると「HISTORY.Xの使用を中止し、現在のファンクションキーの内容を未使用時のキーに登録し、HISTORY.Xの使用を再開する」となる。
／H	編集機能の挿入／上書きの状態を保存する。
／HI	編集機能の挿入／上書きの状態を保存しないで、挿入モードにする（デフォルト）。
／HO	編集機能の挿入／上書きの状態を保存しないで、上書きモードにする。
／I	編集機能を挿入モードにして、そのあと挿入／上書きの状態を保存する。

オプション	内　容
／O	編集機能を上書きモードにして、そのあと挿入／上書きの状態を保存する。
／K	HISTORY.Xの使用を中止する。
／U	／Kで中止されていたHISTORY.Xの使用を再開する。
／L 〈待ち時間〉	簡易バッチ実行時の中断待ち時間を指定する。デフォルトは5で約0.5秒。
／NA	エイリアス機能を使用しない。
／NB	簡易バッチ機能を使用しない。
／NC	ディレクトリ変更履歴機能を使用しない。
／V 〈パス名〉	HISTORY.Xのヘルプファイル（HISTORY.HLP）の所在を指定。
／YA	エイリアス機能を使用する（デフォルト）。
／YB	簡易バッチ機能を使用する（デフォルト）。
／YC	ディレクトリ変更履歴機能を使用する（デフォルト）。
／XA	エイリアス機能を使用する／使用しないを切り替える。
／XB	簡易バッチ機能を使用する／使用しないを切り替える。
／XC	ディレクトリ変更履歴機能を使用する／使用しないを切り替える。

3．組み込み時以外にのみ設定できるもの

オプション	内　　　容
／CA	エイリアス用のバッファをクリア。
／CB	現在の簡易バッチ用バッファをクリア。
／CB〈番号〉	〈番号〉で指定した簡易バッチ用バッファをクリア。
／CBA	すべての簡易バッチ用バッファをクリア。
／CC	ディレクトリ変更履歴用バッファをクリア。
／CH	現在のヒストリ用バッファをクリア。
／CH〈番号〉	〈番号〉で指定したヒストリ用のバッファをクリア。
／CHA	すべてのヒストリ用バッファをクリア。
／C@	すべてのバッファをクリア。
／E	現在のヒストリ用バッファと簡易バッチ用バッファを破棄。
／E〈番号〉	〈番号〉で指定したヒストリ用バッファと簡易バッチ用バッファを破棄。
／EA	すべてのヒストリ用バッファと簡易バッチ用バッファを破棄。
／RA〈ファイル名〉	ファイルの内容をエイリアス用バッファに読み込む。
／RB〈ファイル名〉	ファイルの内容を現在の簡易バッチ用バッファに読み込む。
／RB〈ファイル名〉,〈番号〉	ファイルの内容を〈番号〉で指定した簡易バッチ用バッファに読み込む。 〈番号〉は1から／SHで指定した番号まで指定可能。
／RB〈ファイル名〉,A	ファイルの内容をすべての簡易バッチ用バッファに読み込む。
／RC〈ファイル名〉	ファイルの内容をディレクトリ変更履歴用バッファに読み込む。
／RH〈ファイル名〉	ファイルの内容を現在のヒストリ用バッファに読み込む。
／RH〈ファイル名〉,〈番号〉	ファイルの内容を〈番号〉で指定したヒストリ用バッファに読み込む。〈番号〉は1から／SHで指定した番号まで指定可能。
／RH〈ファイル名〉,A	ファイルの内容をすべてのヒストリ用バッファに読み込む。
／R@〈ファイル名〉	ファイルの内容をすべてのバッファに読み込む。 ／RA、／RB,A、／RC、／RH,Aすべてを指定したときと同じ。
／R+A〈ファイル名〉	ファイルの内容をエイリアス用バッファに追加。
／R+B〈ファイル名〉	ファイルの内容を現在の簡易バッチ用バッファに追加。
／R+B〈ファイル名〉,〈番号〉	ファイルの内容を〈番号〉で指定した簡易バッチ用バッファに追加。 〈番号〉は1から／SHで指定した番号まで指定可能。
／R+B〈ファイル名〉,A	ファイルの内容をすべての簡易バッチ用バッファに追加。
／R+C〈ファイル名〉	ファイルの内容をディレクトリ変更履歴用バッファに追加。
／R+H〈ファイル名〉	ファイルの内容を現在のヒストリ用バッファに追加。
／R+H〈ファイル名〉,〈番号〉	ファイルの内容を〈番号〉で指定したヒストリ用バッファに追加。 〈番号〉は1から／SHで指定した番号まで指定可能。

オプション	内　　容
／R+H〈ファイル名〉，A	ファイルの内容をすべてのヒストリ用バッファに追加。
／R+@〈ファイル名〉	ファイルの内容をすべてのバッファに追加。 ／R+A、／R+B,A、／R+C、／R+H,Aすべてを指定したときと同じ。
／WA〈ファイル名〉	エイリアス用バッファの内容をファイルに書き込む。
／WB〈ファイル名〉	現在の簡易バッチ用バッファの内容をファイルに書き込む。
／WB〈ファイル名〉，〈番号〉	〈番号〉で指定した簡易バッチ用バッファの内容をファイルに書き込む。 〈番号〉は1から／SHで指定した番号まで指定可能。
／WB〈ファイル名〉，A	すべての簡易バッチ用バッファの内容をファイルに書き込む。
／WC〈ファイル名〉	ディレクトリ変更履歴用バッファの内容をファイルに書き込む。
／WH〈ファイル名〉	現在のヒストリ用バッファの内容をファイルに書き込む。
／WH〈ファイル名〉，〈番号〉	〈番号〉で指定したヒストリ用バッファの内容をファイルに書き込む。 〈番号〉は1から／SHで指定した番号まで指定可能。
／WH〈ファイル名〉，A	すべてのヒストリ用バッファの内容をファイルに書き込む。
／W@〈ファイル名〉	すべてのバッファの内容をファイルに書き込む。 ／WA、／WB,A、／WC、／WH,Aすべてを指定したときと同じ。
／W+A〈ファイル名〉	エイリアス用バッファの内容をファイルに追加。
／W+B〈ファイル名〉	現在の簡易バッチ用バッファの内容をファイルに追加。
／W+B〈ファイル名〉，〈番号〉	〈番号〉で指定した簡易バッチ用バッファの内容をファイルに追加。 〈番号〉は1から／SHで指定した番号まで指定可能。
／W+B〈ファイル名〉，A	すべての簡易バッチ用バッファの内容をファイルに追加。
／W+C〈ファイル名〉	ディレクトリ変更履歴用バッファの内容をファイルに追加。
／W+H〈ファイル名〉	現在のヒストリ用バッファの内容をファイルに追加。
／W+H〈ファイル名〉，〈番号〉	〈番号〉で指定したヒストリ用バッファの内容をファイルに追加。 〈番号〉は1から／SHで指定した番号まで指定可能。
／W+H〈ファイル名〉，A	すべてのヒストリ用バッファの内容をファイルに追加。
／W+@〈ファイル名〉	すべてのバッファの内容をファイルに追加。 ／W+A、／W+B,A、／W+C、／W+H,Aすべてを指定したときと同じ。

これまで示したように、HISTORY.Xのオプションは非常にたくさんあります。しかし、そのほとんどはデフォルトのままでも十分ですので、使っていくうちに不便を感じたら、あらためて本書を参照するとよいでしょう。

例1）

```
DEVICE=¥SYS¥HISTORY.X  /D¥HIS¥  /V¥HIS¥  /AR
```

この例は、CONFIG.SYSでデバイスドライバとして組み込むときのものです。HISTORY.Xはサブディレクトリ¥SYSにあるものとして指定しています。オプションの"／D¥HIS¥"は、定義ファイルが¥HISにあるものとして、そのディレクトリの中にある"HISTORY.HIS"と"KEY.HIS"を読み込みます。"／V¥HIS¥"は、ヘルプファイルが¥HISにあるものとして、そのディレクトリの中にある"HISTORY.HLP"を読み込みます。

また、"／AR"はエイリアスの多重定義ができるように指定しています。多重定義というのはエイリアスで定義したコマンドをさらにエイリアスで定義することです。このオプションを付けると、たとえば

```
A>FTREE  TREE  /F  %1■
              ↓ [ESC]・[ I ]
```

と定義しておいて、さらに

```
A>FTPR  FTREE  >  PRN■
              ↓ [ESC]・[ I ]
```

のように定義できます。ここで、

```
A>FTPR  B:¥  [↵]
```

と入力すると

```
TREE  /F  B:¥  >  PRN
```

が実行されます。

例2）

```
HISTORY  /W+@¥HIS¥HISTORY.HIS
```

この例は、HISTORY.Xの使用中に定義した内容を"¥HIS¥HISTORY.HIS"というファイルに追加するものです。電源スイッチを切る前に実行するとよいでしょう。また、これをエイリアスで短い名前にしておくとさらに便利です。

6.4.5　HISTORY.Xの運用について

ここでは、まとめとしてHISTORY.Xを利用していく上でヒントとなるようなことについて触れていきます。

1．エイリアス機能の利用

エイリアス機能の例はこれまでにも示してきましたが、ここでさらに実用的な例をいくつか紹介し

ます。この例を参考にして、ヒストリ定義ファイル（"HISTORY.HIS"）を作成してみてください。

例1）

FMCP　ECHO　%1と%2の準備をしてください　||　PAUSE　||　FORMAT　%2　||　DISKCOPY　%1　%2 ESC・ |

この例は、ディスクのフォーマットとディスクコピーを一括で行うものです。しかし、FORMATコマンドは危険なコマンドなので、ECHOとPAUSEによって確認のメッセージを表示するようにしています。実行するときは、コピー元のドライブ名とフォーマットするドライブ名を指定します。たとえば、ドライブBにあるディスクをフォーマットし、ドライブAからドライブBにコピーするときは、次のように入力します。

FMCP　A：　B：↵

この例のように、IFやGOTOコマンドを使わないものは、バッチファイルを作らなくてもエイリアス機能で代用できます。また、使い方がわからなくなった場合、バッチファイルならばいちいちそのファイルをTYPEコマンドなどで表示して確かめなければなりませんが、エイリアスの場合はその内容を簡単に画面に表示できるという利点もあります。

例2）

BACKALL　ECHO　ドライブBの準備をしてください　||　PAUSE　||BACKUP　A：¥　B：　／S ESC・ |

この例は、ハードディスク（ここではドライブAとしている）のすべての内容をフロッピーディスク（ここではドライブBとしている）にバックアップするものです。ハードディスクを利用しているときは不慮の事故に備えるために定期的にバックアップをとる必要があるので、ときどきこのコマンドでバックアップしてください。実行するときは、

BACKALL ↵

とだけ入力します。

例3）

BACKNEW　ECHO　^[[35mドライブBの準備をしてください^[[m　||PAUSE　||　COPYALL　／T　A：¥DOC　B： ESC・ |

この例は、A：¥DOC以下の新しいファイルだけをドライブBのディスクにコピーするものです。ハードディスク全体のバックアップは例2で行うので、このコマンドでは新しく作ったファイル（ここではA：¥DOC内に新しいファイルを作成するものとしている）だけをバックアップするようにしています。

実行するときは、

BACKNEW ↵

とだけ入力します。

また、この例ではECHOコマンドにエスケープシーケンスを利用して、確認のメッセージを強調しています。エスケープシーケンスというのは、コントロールコードの中のエスケープコード（^[）に続く一連の文字のことで、これによってさまざまな表示のコントロールができます。エスケープコードは、CTRL＋[を押すと入力できます。なお、コントロールコードやエスケープシーケンスについては、付録を参照してください。

例4）

LPFF　ECHO　^L ＞　PRN ESC・I

この例は、プリンタにフォームフィード（紙送り）のコントロールコード（^L）を送るものです。^Lは CTRL＋[を押したあと、CTRL＋L を押します。実行するときは、

LPFF ↵

とだけ入力します。これと同様にしてプリンタのエスケープシーケンスなどを送って、さまざまなプリンタのコントロールができます（プリンタのコントロールコードやエスケープシーケンスはプリンタのマニュアルを参照してください）。

例5）

SAVHIS　HISTORY　／W＋@￥HIS￥HISTORY.HIS ESC・I

この例は前にも示したものですが、HISTORY.Xを使用中に定義したエイリアスやヒストリなどをファイルに追加して書き込むものです。実行するときは、

SAVHIS ↵

とだけ入力します。

例6）

RSINIT　SPEED　1200　B8　PN　S1　XON ESC・I

この例も前に示したものですが、このようにオプションの多いコマンドにはエイリアス機能はたいへんに便利です。

例7）

APATH　PATH　%path%；%1 ESC・I

この例は、すでに設定してあるコマンドパスにさらにパスを追加するためのものです。たとえば、A：￥UTLを追加したいときは、

APATH A：¥UTL ⏎

と入力します。

ここではエイリアス以外の例は示しませんでしたが、ディレクトリ変更履歴には普段よく使うディレクトリを、ヒストリにはよく使うコマンドをヒストリ定義ファイル（"HISTORY.HIS"）に登録しておくとよいでしょう。

2. ヘルプファイルについて

HISTORY.Xには便利な機能がたくさんありますが、機能が多すぎて一度にはなかなか憶えきれないものです。だからといっていちいち本書を開くのもめんどうです。そこで、システムディスクにはHISTORY.Xの使い方を表示するためにヘルプファイルが用意されています。

ヘルプファイルのファイル名は"HISTORY.HLP"として、¥HISのディレクトリの中にあります。このヘルプファイルは組み込み時にスイッチ／Vで、そのディレクトリを指定します。

例）

DEVICE＝¥SYS¥HISTORY.X ／D¥HIS¥ ／V¥HIS¥ ／AR

この例では定義ファイルやヘルプファイルが"¥HIS¥"にあるものとして指定していますが、このように必ず最後に¥記号をつけてください。スイッチ／Vの指定がない場合、スイッチ／Dで指定したディレクトリにヘルプファイルがあるものとして、それを読み込みます。ただし、スイッチ／Dで指定したディレクトリにヘルプファイルがない場合は、HISTORY.Xのあるディレクトリからヘルプファイルを読み込みます。

ヘルプファイルを表示するときは、HELP キーを押してください。「ヒストリーヘルプメニュー」が表示されますので、メニューの指示にしたがってください。

6．5　キーボードコントロール機能

キーボードコントロール機能は2つのキーを組み合わせて使用することにより、画面やドライブなどをコントロールする機能です。CTRL キーを押しながらもう1つのキーを押すことにより実行します。
次にキーボードコントロール機能の一覧を示します。

キーボードコントロール機能

キー	機　能
CTRL + F 1	ドライブAのイジェクトまたはハードディスクのOFF
CTRL + F 2	ドライブBのイジェクトまたはハードディスクのOFF
CTRL + F 3	ドライブCのイジェクトまたはハードディスクのOFF
CTRL + F 4	ドライブDのイジェクトまたはハードディスクのOFF
CTRL + F 7	ファンクションキーの表示の変更
CTRL + OPT.1 + DEL	システムのリセット
CTRL + C	実行中のコマンドの中止
CTRL + H	コマンド行から最後の文字を削除（BS と同様）
CTRL + P	画面表示をプリンタにも出力（一度押すとプリンタ出力の設定、もう一度押すと出力の解除）
CTRL + N	プリンタへの出力を解除
CTRL + S	画面表示、スクロールの一時停止。任意キーで解除

注意

CONFIG.SYSにヒストリデバイスドライバHISTORY.Xが組み込まれているか、コマンド行からHISTORY.Xを使用できるようにした場合、CTRL+P 、CTRL+Nが使えません。
CTRL+P 、CTRL+Nを使う場合は、HISTORY　／K　⏎を実行してください。

第7章　システムの構築

7．1　イントロダクション

Human68k は、システム自体が非常に柔軟で拡張性に富んでいます。使用する状況に応じて、起動時に必要なデバイスドライバをシステムに付け加えたり、そのデバイスドライバの機能をあらかじめ設定しておいたり、というような、さまざまな要求に答えられるように設計されています。なお、デバイスドライバはシステムディスク上の〝SYS〟というディレクトリの中にあります。

この章では、こうした数々のシステム構成を作りあげる方法について解説します。

7．2　CONFIG.SYS ファイルのコマンド

Human68k では、システムが起動するときに、ディスクファイルの中から CONFIG.SYS と名付けられたファイルをさがし、その内容により、システムが作動する環境を設定するようになっています。

このCONFIG.SYSファイルは、「システム構築用ファイル」と呼ばれ、以降で説明する各種のコマンドが文字列で記述された、テキストファイルとなっています。

したがって、CONFIG.SYS ファイルを書きかえることにより、自分の目的に応じたシステムの環境を作り出すことができます。

CONFIG.SYS ファイルはテキストファイルですから、その内容はスクリーンエディタ ED や、SX-WINDOW の日本語マルチフォントエディタなどで作成・編集することができます。

CONFIG.SYS ファイルの内容設定に使用できるコマンドは次のとおりです。

コマンド	機　　能
BELL	ビープ音に使用するPCMファイルの指定
BREAK	CTRL + C のチェック機能の設定
BUFFERS	ディスクバッファの数とバッファ容量の設定
COMMON	COMMONファンクションで使用するメモリ容量の指定
DEVICE	デバイスドライバをシステムリストに登録
ENVSET	環境エリアの確保と環境ファイルのセット
FILES	ファンクションコールで同時にオープンできるファイルの数を指定
KEY	キー定義ファイルの指定
LASTDRIVE	仮想ドライブを含めた最大のドライブ数を指定
PROCESS	並行処理のための制御情報の設定
PROGRAM	シェル起動の前に実行するプログラムの指定
SHARE	ファイルの共有と排他制御の指定
SHELL	指定したファイルをシェル（コマンドプロセッサ）として実行
TITLE	タイトルファイルの指定
USKCG	外字登録用データファイルの指定
VERIFY	ベリファイ機能の設定

各コマンドの詳細については、以降のコマンド別解説を参照してください。

●CONFIG.SYSの例

CONFIG.SYSの例を次に示します。

```
FILES       = 15
BUFFERS     = 20  1024
LASTDRIVE   = Z:
KEY         = ¥KEY.SYS
USKCG       = ¥USKCG.SYS
BELL        = ¥BEEP.SYS
DEVICE      = ¥SYS¥PRNDRV.SYS
DEVICE      = ¥SYS¥ASK68K.SYS  /DB:¥X68K.DIC  /E¥ASK¥ENV1.ASK
DEVICE      = ¥SYS¥RSDRV.SYS
DEVICE      = ¥SYS¥OPMDRV3.X
DEVICE      = ¥SYS¥FLOAT2.X
DEVICE      = ¥SYS¥HISTORY.X  /D¥HIS¥  /SH2,8,4
DEVICE      = ¥SYS¥IOCS.X
ENVSET      = 512  ¥STARTUP.ENV
```

このようなCONFIG.SYSでHuman68kを起動すると、次のように設定されます。

●FILESは15、BUFFERSはバッファ数20、バッファ容量1024バイトに設定
●最終のドライブをドライブZに設定
●KEY.SYS、USKCG.SYS、BEEP.SYSをルートディレクトリから読み込む
●プリンタ(PRNDRV.SYS)、日本語FP(ASK68K.SYS)、RS-232C(RSDRV.SYS)、FM音源・ADPCM・MIDI(OPMDRV3.X)、数値演算(FLOAT2.X) のデバイスドライバを登録
●ヒストリデバイスドライバHISTORY.Xによるヒストリ機能などを使用可能に設定
ヒストリ定義ファイル、キー定義ファイルをＨＩＳディレクトリから読み込み、ヒストリ用バッファを2個（8KBと4KB）確保
●グラフィック描画、テキスト表示を高速化するIOCS.Xを登録
●環境エリアとして512バイトを確保し、環境ファイルSTARTUP.ENVの内容を設定

BELL

機　能　ビープ音に使用する PCM ファイルの指定

書　式　BELL＝〔〈d：〉〕〔〈パス名〉〕〈ファイル名〉

解　説

Human68k システムでは、エラーの発生時や、コンソールに対して CTRL ＋ G を送ったときなどに、ビープ音が鳴ります。BELL は、このビープ音を発生するためのデータファイルを指定するものです。

〈ファイル名〉には、鳴らすべき音のデータを ADPCM 方式で記録したファイルを指定し、〈d：〉〈パス名〉にはそのファイルの所在を指定します。指定可能なファイルの大きさは64K バイト以内です。64K バイト以上のファイルを指定した場合は、64K バイト分のデータのみが有効となり、残りのデータは無視されます。

BELL を指定しなかったときは、コンソールに対して CTRL ＋ G を送っても、またエラーの発生時にも、ビープ音は鳴りません。

BREAK

機　能　CTRL ＋ C のチェック機能の設定

書　式　BREAK＝〔ON | OFF | KILL〕

解　説

ふつう、CTRL ＋ C をキーボードから入力することにより、実行中のコマンドやプログラムを中止することができます。

BREAK は、この CTRL ＋ C による割り込みをチェックする機能を設定するために使用します。

BREAK＝OFF に設定すると、コンソール入出力時とプリンタ出力時にのみ CTRL ＋ C が有効となります。プログラムがこれ以外の作業を行っている場合は、CTRL ＋ C を押しても、プログラムの実行を中止することはできません。

BREAK＝ON に設定すると、すべての場合に、CTRL ＋ C が有効となります。したがって、プログラムのコンパイルを行っているときなどのように、ディスクをアクセスしているときにも、CTRL ＋ C を押すことによりプログラムの実行を中止することができます。

BREAK＝KILL に設定すると、どのような場合も実行を中止することはできません。どうしても中止したいときは、本体の INTERRUPT スイッチを押してください。

なお、この CONFIG. SYS ファイル用の BREAK コマンドは、Human68k コマンドモード中で使用できる BREAK コマンドと、その働きは同じです。第5章の BREAK コマンドを参照してください。

BUFFERS

機　能　起動時に Human68k が割り当てるディスクバッファ数とバッファ容量を指定

書　式　BUFFERS=〈ディスクバッファ数〉〔〈バッファ容量〉〕

解　説

ディスクとメモリ間などでやりとりされるデータは、一時的にディスクバッファと呼ばれる部分に保存されます。BUFFERS は、このディスクバッファの数を指定するもので、これを大きくするとディスクアクセスの速度が向上します。

〈ディスクバッファ数〉で指定できる範囲は2から99です。無指定時のデフォルト値は20です。

日本語ワードプロセッサやデータベースなど、大量のデータをディスクと頻繁にやりとりするようなアプリケーションプログラムを実行するときなどは、ディスクバッファ数を増加させるとよいでしょう。サブディレクトリの数が非常に多い場合なども、このディスクバッファ数を増加させると効果的です。

バッファ容量とは、1つのディスクバッファに割り当てるメモリの容量で、バイト数で指定します。指定できる範囲は1024～32768で、指定しない場合は1024となります。ディスクバッファはデバイスドライバから1セクタずつ読み込んで保存しておくためのバッファなので、1セクタが1024バイト以上のデバイスドライバを組み込むときは、その最大のバイト数を指定する必要があります。

この指定によって確保されるメモリは、次のような計算式で算出できます。

ディスクバッファ数×（バッファ容量+16）バイト

COMMON

機　能　COMMON ファンクションで使用するメモリ容量の指定

書　式　COMMON=〈容量〉

解　説

各アプリケーションは、COMMONファンクションを利用して内部のコモンエリアをアクセスすることができます。コモンエリアとは、Human68kのシステムエリアに設けられた領域です。このコモンエリアを使うと、たとえば複数のプロセスでこのコモンエリアを介してデータの引き渡しができるようになります。COMMONは、このコモンエリアの容量をKバイト単位で指定するものです。指定しないと、COMMONファンクションは利用できず、エラーを返します。

DEVICE

機　能	ファイル名で指定したデバイスドライバの、システムへの登録
書　式	DEVICE=〔<d：>〕〔<パス名>〕<ファイル名>　〔<オプション>〕

解　説

DEVICEは、デバイスドライバをHuman68kシステムに登録する場合に使用します。<ファイル名>には、デバイスドライバの様式で作成されたプログラムのファイル名を記載し、<d：> <パス名> には、そのプログラムの所在を指定します。

そのデバイスドライバがオプションを指定できるように作られたものであれば、<オプション> に、指定したいオプションを記載します。

デバイスドライバをCONFIG.SYSファイル中に指定しておくと、Human68k起動時にそのデバイスドライバがシステムに読み込まれてメモリ上に常駐します。

以後、設定したデバイスを使用することができるようになります。

例

```
DEVICE=¥SYS¥PRNDRV.SYS
```

上記のように指定すると、プリンタを使用できるようにシステムに設定されます。これが書かれていなければ、プリンタは使えないことになります。

<デバイスドライバ>

ASK68K.SYS

日本語ＦＰを使用するためのデバイスドライバです。オプションに、使用する辞書名と環境ファイル名を指定します。

詳しくは、「日本語入力・辞書ユーティリティユーザーズマニュアル」を参照してください。

書式例　DEVICE = ¥SYS¥ASK68K.SYS　/DB：¥X68K.DIC　/E¥ASK¥ENV1.ASK

FDDEVICE.X

2HD以外のフロッピーディスクを読み書きするためのデバイスドライバです。詳しくは、本書「7.3　FDDEVICEドライバについて」を参照してください。

書式例　DEVICE=¥SYS¥FDDEVICE.X

FLOAT2.X

数値演算を使用するためのデバイスドライバです。浮動小数点のデータフォーマットはIEEEフォーマットです。IEEEフォーマットとは、別売の「C Compiler PRO-68k」などで使用されているデータフォーマットです。X68030シリーズでは、本体同梱のバージョン以降のものをお使いください。

書式例　DEVICE=¥SYS¥FLOAT2.X

FLOAT3.X

数値演算プロセッサボードCZ-6BP1を使用するためのデバイスドライバです。浮動小数点のデータフォーマットはIEEEフォーマットです。X68030シリーズでは、本体同梱のバージョン以降のものをお使いください。

書式例　DEVICE=¥SYS¥FLOAT3.X

FLOAT4.X

数値演算プロセッサCZ-5MP1を使用するためのデバイスドライバです。
浮動小数点のデータフォーマットはIEEEフォーマットです。

書式例　DEVICE=¥SYS¥FLOAT4.X

HISTORY.X

ヒストリデバイスドライバによる、ヒストリ機能、エイリアス機能、簡易バッチ機能などを使用するためのデバイスドライバです。詳しくは、第6章「6．4 ヒストリデバイスドライバ」を参照してください。

書式例　DEVICE=¥SYS¥HISTORY.X　／D¥HIS¥　／SH 2，8，4

OPMDRV3.X

FM音源やADPCM、MIDIを使用するためのデバイスドライバです。
詳しくは、本書「7．4 OPMドライバについて」を参照してください。

書式例　DEVICE=¥SYS¥OPMDRV3.X　＃120　／MOP　／Y0

PRNDRV.SYS

プリンタを使用するためのデバイスドライバです。

詳しくは、本書「7．5プリンタの設定について」を参照してください。

書式例　DEVICE=¥SYS¥PRNDRV.SYS

RAMDISK.SYS

グラフィックRAMやメインメモリを、RAMディスクとして使用するためのデバイスドライバです。オプションにメモリの種類と容量を指定します。

詳しくは、本書「7．6RAMディスクについて」を参照してください。

書式例　DEVICE=¥SYS¥RAMDISK.SYS　#GM128

RSDRV.SYS

増設用RS-232Cボードを使用するためのデバイスドライバです。

書式例　DEVICE=¥SYS¥RSDRV.SYS

SRAMDISK.SYS

スタティックRAM（内蔵のバッテリバックアップメモリ）を、RAMディスクとして使用するためのデバイスドライバです。詳しくは、本書「7．6RAMディスクについて」を参照してください。

書式例　DEVICE=¥SYS¥SRAMDISK.SYS

DIRSCH

機 能　ファイル検索効率の向上

書 式　DIRSCH=［ON｜OFF］

解 説　DIRSCHは、ファイルの検索効率を向上させます。通常のファイル検索では、ファイル検索ごとにはじめから検索を行いますが、DIRSCH=ONと設定すると、ファイルを見つけた場所から続けて検索を行いますので、順番に並んでいるファイルを続けて検索するときなどに、検索効率が大幅に向上します。

ENVSET

機 能　環境エリアの確保と環境ファイルのセット

書 式　ENVSET=〈環境エリア容量〉［［〈d：〉］［〈パス名〉］〈ファイル名〉］

解 説　環境エリアの確保と環境ファイルに設定された内容にしたがって環境変数のセットを行います。CONFIG.SYSの"SHELL"起動の前に"PROGRAM"でプログラムを起動する場合やコマンドプロセッサとしてCOMMAND.X以外（たとえばBASIC.Xなど）を起動する場合に使用します。

〈環境エリア容量〉は、バイト単位で指定します。〈ファイル名〉には、環境ファイルを、〈d：〉〈パス名〉には、そのファイルの所在を指定します。標準では"STARTUP.ENV"が指定されています。

〈ファイル名〉を省略した場合は、環境エリアのみ確保され、環境ファイルはセットされません。環境ファイルとは、左辺に"環境変数名="、右辺にその内容を列記したものです。

環境ファイルの例

```
path=A:¥;A:¥BIN;A:¥UTL
WORK=A:¥DOC
HELP=A:¥HELP
```

EXCONFIG（CONFIGED）

機　能　CONFIGEDの登録によるCONFIG.SYSの選択や修正

書　式　EXCONFIG=CONFIGED〔<オプション>〕

解　説　CONFIGEDは、CONFIG.SYSにEXCONFIG=で登録して使用するためのプログラムであり、CONFIG.SYSの選択機能と修正機能があります。

書式例　EXCONFIG=CONFIGED -KSHIFT

-K(KEY)　選択画面に移行するときに押すキー名を登録します
ここで登録できるキーはSHIFT、CTRL、OPT1、OPT2、の４種類のみです
省略すると、SHIFTが登録されます

CONFIG.SYSに上記のように登録し、SHIFTを押しながら起動すると、次のように表示されます。

```
          CONFIGEDIT Version 1.00

     A:¥      CONFIG.SYS 1011 93-02-15 12:00:00
     A:¥SX¥   CONFIG.SYS  566 93-02-20 12:00:00
     A:¥CONF¥CONFIG.TMP   333 93-03-11 12:00:00
     A:¥      CONFIG.SYS  820 93-02-23 12:00:00
     A:¥USR¥  CONFIG.TMP  645 93-02-20 12:00:00

 [↑↓]:選択  [ENTER]:確定 [ESC]:取消
```

システムを起動したドライブ（複数領域を確保しているときは起動した領域）の２階層目までのディレクトリにある、すべてのCONFIG.SYSならびに、CONFIG.TMPが表示されます。カーソルキーの↑↓でどのCONFIGファイルを登録、または編集するかを選択し↵押すと、CONFIGデータの修正画面が表示されます。

CONFIGファイルが、1つしかないときは、直接CONFIGデータの修正画面が表示されます。

```
          CONFIGEDIT Version 1.00 Edit File=CONFIG.SYS

 ○  FILES      = 15
 ○  BUFFERS    = 20 1024
 ○  LASTDRIVE  = Z:
 ○  EXCONFIG   = CONFIGED -KSHIFT
 ×  KEY        = ¥KEY.SYS
 ○  USKCG      = ¥USKCG.SYS
 ○  BELL       = ¥BEEP.SYS
 ×  DEVICE     = ¥SYS¥PRNDRV.SYS
 ×  DEVICE     = ¥SYS¥RSDRV.SYS
 ○  DEVICE     = ¥SYS¥OPMDRV3.SYS
 ○  DEVICE     = ¥SYS¥FLOAT2.X
 ○  DEVICE     = ¥SYS¥ASK68K.SYS /DB:¥X68K.DIC /E¥ASK¥ENV1.ASK
 ○  DEVICE     = ¥SYS¥HISTORY.X /D¥HIS¥ /SH2,8,4
 ○  ENVSET     = ¥512 ¥STARTUP.ENV



[↑↓]:選択 [SPACE]:有効(○)/無効(×)切替 [→]:修正 [ENTER]:実行 [ESC]:取消
```

カーソルキーの [↑][↓] で修正する項目を選択します。
選択された状態で [スペース] を押すと、その項目を有効にするか無効にするかを設定することができます。

[↵] を押すと、修正したCONFIGファイルを、CONFIG.TMPというファイル名でルートディレクトリーに保存し、修正した内容で起動します。以後、修正した内容で起動するときは、選択画面に移行するためのキーを押しながら起動し、CONFIG.TMPを選択してください。

[ESC] を押すと、選択されているCONFIGファイルの修正前の内容で起動します。

選択された状態で [→] を押すと、その項目の内容を編集することができます。
[→] を押すと、次のように表示されます。

```
        CONFIGEDIT Version 1.00 Edit File=CONFIG.SYS

○ FILES     = 15
○ BUFFERS   = 20 1024
○ LASTDRIVE = Z:
○ EXCONFIG  = CONFIGED -KSHIFT
× KEY       = ¥KEY.SYS
○ USKCG     = ¥USKCG.SYS
○ BELL      = ¥BEEP.SYS
× DEVICE    = ¥SYS¥PRNDRV.SYS
× DEVICE    = ¥SYS¥RSDRV.SYS
○ DEVICE    = ¥SYS¥OPMDRV3.SYS
○ DEVICE    = ¥SYS¥FLOAT2.X
○ DEVICE    = ¥SYS¥ASK68K.SYS /DB:¥X68K.DIC /E¥ASK¥ENV1.ASK
○ DEVICE    = ¥SYS¥HISTORY.X /D¥HIS¥ /SH2,8,4
○ ENVSET    = ¥512 ¥STARTUP.ENV



[←→]:移動 [DEL/BS/CLR/CTRL+K]:削除 [INS]:挿入モード [ENTER]:決定 [ESC]:取消
```

キー	機能
[←]、[→]	カーソルを移動します
[DEL]	カーソル位置の１文字を削除します
[BS]	カーソル位置の前の１文字を削除します
[CLR]	入力エリア全体を消去します
[CTRL] + [K]	カーソル位置から後ろをすべて削除します
[INS]	挿入モードのON／OFFを設定します
[↵]	修正内容を決定して修正項目選択画面に戻ります
[ESC]	もとの内容に戻り、修正項目選択画面に戻ります

ご注意

この修正では、漢字の修正および入力やローマ字入力、コード入力などを行うことができません。

FILES

機能　ファンクションコールで同時オープンできるファイル数の指定

書式　FILES=〈ファイル数〉

解説

FILESは、ファンクションコールで同時オープンできるファイルハンドラの数を、〈ファイル数〉で指定された数に設定します。

〈ファイル数〉で指定できる範囲は5から93です。無指定時のデフォルト値は15です。

```
FILES=20
```

とすると、ファンクションコールで同時に20のファイルをオープンすることが可能となります。

なお、〈ファイル数〉には、必ず5以上を指定するようにしてください。

ファンクションコールの詳細については、別売の「C Compiler PRO-68K」で説明しています。

KEY

機能　キー定義ファイルの指定

書式　KEY=〔〈d：〉〕〔〈パス名〉〕〈ファイル名〉

解説

KEYコマンドで作成されたキー定義ファイルとその所在を指定します。この"KEY"が省略された場合、起動ドライブのルートディレクトリに"KEY.SYS"があればその内容が設定され、"KEY.SYS"がなければデフォルトのキー定義が設定されます。

LASTDRIVE

機　能　仮想ドライブを含めた最大のドライブ数を指定

書　式　LASTDRIVE=〈d：〉

解　説　SUBST コマンドやネットワーク環境中で使用する仮想ドライブの最大値を設定します。〈d：〉で指定できるドライブ名はA～Zです。ただし、実際に接続されているドライブ数より小さい場合やパラメータエラーの場合は、実際に接続されている最終のドライブ名になります。また、"LASTDRIVE"の指定がない場合は、ドライブZとなります。

PROCESS

機　能　並行処理のための制御情報の設定

書　式　PROCESS=〈プログラム数〉〈レベル〉〈タイムスライス値〉

解　説　並行処理（複数のプログラムを見かけ上同時に動かすこと）を行うプログラム数などを設定します。〈プログラム数〉には並行実行可能なプログラムの数を指定します。指定可能な範囲は2～32です。

〈レベル〉にはプログラムの実行間隔を指定します。指定できる値は2～255で、値が小さいほど短い間隔で実行されます(実行速度が速くなる)。

〈タイムスライス値〉には各プログラムの実行時間を指定します。指定できる値はミリ秒単位で1～100の値です。（1ms～100ms)

"PROCESS"の指定がない場合、並行処理はできません。なお、並行処理の状態はPROCESSコマンドで表示することができます。

PROGRAM

機 能　シェル起動の前に実行するプログラムの指定

書 式　PROGRAM=〔<d:>〕〔<パス名>〕<プログラム名>〔オプション〕

解 説　CONFIG.SYSの"SHELL"によって、シェル（コマンドプロセッサ）が起動されますが、この"PROGRAM"にはそのシェルの前に実行するプログラムを指定します。これはメモリに常駐するプログラムやシェルがCOMMAND.Xでない場合（たとえばVS.Xなど）に実行するプログラムを起動するためのものです。ただし、指定できるプログラムはSUBSTコマンドなどの外部コマンドに限られ、COPYコマンドなどの内部コマンドはCOMMAND.Xの機能のひとつなので指定することはできません。また、"PROGRAM"を使用する場合は、あらかじめ"ENVSET"で環境エリアを確保しておいてください。

SCSIDEV

機 能　SCSI仕様のデバイスを使用可否の設定

書 式　SCSIDEV=〔ON｜OFF〕

解 説　SCSIDEVは、SCSIデバイスドライバを登録するのかどうかを設定するために使用します。SCSIデバイスを使用されない方はSCSIDEV=OFFに設定すると、起動が速くなります。

SCSIDEV=ONに設定すると、SCSIROMのデバイスドライバを登録します。
SCSIDEV=OFFに設定すると、SCSIデバイスドライバは登録しません。

SHARE

機　能	ファイルの共有と排他制御の指定
書　式	SHARE= 〈ファイル数〉〈ロック領域数〉
解　説	ネットワークを利用しているときは、複数のユーザーが1つのファイルを同時にアクセスする可能性があります。このような場合、ファイルの読み出しおよび書き込みをすべてのユーザーに許すと、ファイルの整合性がとれなくなるようなことが発生してしまいます。このような不整合を防ぐために、1人のユーザーがあるファイルをアクセスしているときは、そのファイルを他のユーザーがアクセスできなくする（これを排他制御機能あるいはロック機能と呼びます）ようにしなければなりません。これを行うように指定するのがこの〝SHARE〟です。 〈ファイル数〉には同時に管理するファイル数を、〈ロック領域数〉にはそれぞれのロック領域の数を指定します。

SHELL

機　能	起動時のシェル(コマンドプロセッサ)の指定
書　式	SHELL=〔〈d：〉〕〔〈パス名〉〕〈ファイル名〉〔〈オプション〉〕
解　説	SHELLは、コマンドプロセッサを指定するときに使用します。この指定を行うと、システムを起動したときに指定したファイルが読み込まれ、コマンドプロセッサとして実行されます。 〈ファイル名〉には、実行可能な〝.X〟または〝.R〟の拡張子をもつプログラムを、〈d：〉〈パス名〉にはそのファイルの所在を指定します。そのプログラムがオプションの指定ができるように作成されていれば、〈オプション〉に指定したいオプションを添えます。 SHELLを指定しないとき、デフォルトのコマンドプロセッサは、COMMAND.Xとなります。

TITLE

機　能　タイトルファイルの指定

書　式　TITLE=〔〈d：〉〕〔〈パス名〉〕〈ファイル名〉

解　説　タイトルファイルのファイル名とその所在を指定します。"TITLE" が省略された場合、起動ドライブのルートディレクトリに "TITLE.SYS" があればその内容をタイトルとして表示し、なければ表示しません。

USKCG

機　能　外字登録用データファイルの指定

書　式　USKCG=〔〈d：〉〕〔〈パス名〉〕〈ファイル名〉

解　説　外字登録用データファイルのファイル名とその所在を指定します。"USKCG" が省略された場合、起動ドライブのルートディレクトリに "USKCG.SYS" があればその内容を外字登録文字として定義し、なければ定義しません。

VERIFY

機　能　ベリファイ機能の設定

書　式　VERIFY=〔ON | OFF〕

解　説　VERIFYは、ディスクに書き込みを行う際に、正常に書き込めたかどうかのチェックを行うか否かの設定をします。

VERIFY=ONに設定すると、書き込みの直後に、書き込んだデータを読み出してそのデータが正しく書けたかどうかの照合チェックを行います。したがって、不良セクタなどのために物理的に書き込めない部分があるときは、即座にエラーとして表示されるようになります。

ただし、VERIFY=ONに設定されている場合、照合チェックを行いながら書き込むため、ディスクに書き込みを行う速度は遅くなります。

VERIFYをCONFIG.　SYSファイル中にとくに指定しない場合、VERIFY=OFFとなります。

なお、このCONFIG.SYSファイル用のVERIFYコマンドは、Human68kコマンドモード中で使用できるVERIFYコマンドと、その働きは同じです。第5章のVERIFYコマンドを参照してください。

7.3 FDDEVICEドライバについて

このプログラムは、X68000／X68030シリーズのIOCSコールを、2HD以外のディスクの読み書きを許すように変更します。

このプログラムは常駐することでこれらのサポートを行い、常駐している間は、OSコールでフロッピーディスクをアクセスする限り、以下のメディアを自動的に識別して、アクセスを可能にします。ただし、この機能を使うためには2DD/2HD共用のドライブが必要です。

自動判別するメディア

●5インチフロッピーディスクドライブ

2HD	1024バイト×8セクタ×2面×77トラック	X68030／PC9801 2HD
2HC	512バイト×15セクタ×2面×77トラック	IBM 2HC

●3.5インチフロッピーディスクドライブ

2HD	1024バイト×8セクタ×2面×77トラック	X68030／PC9801 2HD
2HC	512バイト×15セクタ×2面×77トラック	IBM 2HC
2DD	512バイト×8セクタ×2面×80トラック	PC9801 2DD (CZ-300C／310Cのみ)
2DD	512バイト×9セクタ×2面×80トラック	IBM 2DD (CZ-300C／310Cのみ)

※FASTIO／FASTSEEK／FASTOPENを実行する前にFDDEVICEを登録してください。

7.4　OPMドライバについて

本体内蔵のFM音源やADPCM、またMIDI関連の機器を使用するためには、CONFIG.SYSファイル中で、DEVICEコマンドによってOPMDRV3.Xを登録する必要があります。

●サポートデバイス名

OPMDRV3.Xは、以下のデバイス名をサポートしています。

PCM　：PCMの録音／再生

MIDI、OPM：MMLデータの再生、および各コマンドの実行

MIDIA　：アスキーファイルをMIDIコードに変換してMIDI OUTへ出力

MIDIB　：バイナリーファイルをそのままMIDIデータとして出力

MIDIAE　：アスキー　エクスクルーシブデータ　ファイルをMIDIコードに変換して出力
(Musicstudio PRO-68Kで作成されるEXCファイルなど)

MIDIBE　：バイナリー　エクスクルーシブデータ　ファイルをそのままMIDIデータとして出力
(MU－1で作成されるEXBファイルなど)

ご注意

・エクスクルーシブデータを出力するときのデバイス名は、'MIDIAE' または 'MIDIBE' で行ってください。
('F0' 'F7' のチェックを行い、受信側がオーバーフローをおこさないようにウェイトを挿入します。)

※「MU－1」は株式会社サンミュージカルサービスの商標です。

●オプションスイッチ

OPMDRV3.Xでは登録時に以下のオプションスイッチを設定できます。

#n…………nにトラックバッファのサイズを1Kバイト単位で指定します。
指定範囲は24Kバイト～4096Kバイトで、省略した場合は64Kバイトを確保します。

書式例　DEVICE=¥SYS¥OPMDRV3.X　#120

／Pn……… nにPCMデータのバッファサイズをKバイト単位で指定します。
指定範囲は0Kバイト～4096Kバイトで、省略したときは24Kバイトを確保します。
ADPCMを使用しないときは、nに0を指定してください。

書式例　DEVICE=¥SYS¥OPMDRV3.X　／P200

／Yn……… MMLデータの Yコマンドのモードを指定します。デフォルトは／Y1に設定されます。

書式例　DEVICE=¥SYS¥OPMDRV3.X　／Y1

／OMP…… アルファベットで指定された順に、各音源で使用するチャンネル番号を割り当てます。

O=OPM　(使用チャンネル数8　：ch＝1～8)
M=MIDI　(使用チャンネル数16　：ch＝9～24)
P=PCM　(使用チャンネル数1　：ch＝25)

／OMPとした場合は、それぞれの音源に対するチャンネル番号が、上記の通りとなります。
省略した場合は、／OPMの設定になり、次の通りとなります。

O=OPM　(使用チャンネル数8　：ch＝1～8)
P=PCM　(使用チャンネル数1　：ch＝9)
M=MIDI　(使用チャンネル数16　：ch＝10～25)

MIDIを使用しない場合は、／OPMの指定にしておけば、従来のMMLデータをそのまま演奏できます。

MIDIに指定されたchは、その範囲をMIDIch1～16にシフトしてMIDIOUTに出力されます。'／OPM' でch10～25が指定されている場合は、ch10がMIDIch1に、ch25がMIDIch16になります。

書式例　DEVICE=¥SYS¥OPMDRV3.X　／OMP

※詳しくは、本項「OPMコマンド」または、『X-BASICユーザーズリファレンスマニュアル』" m_sysch () コマンド" を参照してください。

／Fn………FM音源全体の音程を微調整します。(n＝0～63) デフォルトは0

書式例　DEVICE=¥SYS¥OPMDRV3.X　／F12

／R…………OPMDRV3.Xを解除します。

書式例　DEVICE=¥SYS¥OPMDRV3.X　／R

／A…………MUSIC PRO-68K［MIDI］を使用するときに設定します。

書式例　DEVICE=¥SYS¥OPMDRV3.X　／A

参考

書式例　トラックバッファに120Kバイトを、PCMデータのバッファに200Kバイトを確保し、MUSIC PRO-68K［MIDI］のMUSファイルを演奏する場合

DEVICE=¥SYS¥OPMDRV3.X　♯120　／P200　／Y0　／MOP

※各スイッチを省略した場合のデフォルトは、♯64　／P24　／Y1　／OPMとなります。

※それぞれの曲データの先頭に以下のOPMコマンドを追加記入しておくと、上記の選択は不要になります（本項「7.5.1　OPMコマンド」を参照)。

OPMファイルを演奏する場合（／OPM）（／Y1）
MUSファイルを演奏する場合（／MOP）（／Y0）

※OPMDRV3.Xと各スイッチの設定は、起動後、コマンドモードからも実行できます。

ご注意

・OPMDRV3.XはOPMDRV.Xの上位互換になっていますが、OPMDRV2.Xとは互換性がありませんのでご注意ください。

7.4.1　音源の制御

本機は YM2151という FM 音源 LSI を内蔵しています。これは、一般に OPM と呼ばれているタイプで、市販のシンセサイザにも搭載されているたいへん高機能なものです。OPM は8オクターブの FM 音源を8チャンネル内蔵し、各チャンネルごとに音色を設定したり、左／右あるいはその両方に出力を振り分けることができます。つまり、これひとつで8オクターブ8重和音のステレオ演奏が可能となるわけです。

本機の内蔵スピーカはモノラルですが、本体後面にはオーディオ出力端子が装備されていますので、これをオーディオアンプやラジカセなどに接続することにより、ハイファイステレオサウンドを楽しむことができます。

FM 音源は発振器（これをオペレータといいます）で周波数変調（Frequency Modulation）をかけることによって音色を作り出します。OPM はチャンネルごとに4つのオペレータを持っていますので、その組み合わせ方（アルゴリズム）や各オペレータの状態を操作することで、無数ともいえる音色を出すことができます。

演奏には、実際の楽譜に対応したMMLという言語を使います。このMMLで記述された音色や音符などの楽譜データをすべてトラックバッファに書きこみます。トラックバッファはFM音源の各チャンネルに固定されているのではなく、必要に応じて領域を確保し、自由に各チャンネルに割りあてることができます。また、演奏時にはチャンネルを指定することができます。これによって、より効率的な楽譜データの記述や多彩な音楽演奏が可能となります。

FM 音源は、一般に次の手順で使用します。

1　FM 音源を初期化する
2　MML データ(文字列)を格納するトラックバッファを確保する
3　トラックバッファを FM 音源の各チャンネルに割りあてる
4　演奏のテンポを設定する
5　トラックバッファに MML データを格納する
6　演奏する

なお、総トラックバッファサイズ（それぞれのトラックに割りあてられているトラックバッファの総合計）を変更する場合は、CONFIG.SYSファイル内のOPMDRV3.Xの行を次のように書きかえてから、再度システム起動してください。

DEVICE = ¥SYS¥OPMDRV3.X　♯100

総トラックバッファのサイズ（単位：KB）

この例では、システム再起動後、約100KBの総トラックバッファが確保されます。何も指定しない場合は、約64KBの総トラックバッファが確保されます。ただし、システム起動時のメモリのフリーエリアはこのトラックバッファサイズに関係してきますので注意してください。

●MMLデータ

FM音源やMIDI音源に対して楽譜データを出力するときに使用する書式がMMLです。

まず、MMLと楽譜を対応させてみましょう。

楽譜

MML

```
準備            1小節目              2小節目          3小節目           4小節目
o4v9q7l16 |:   v10T100@16>b2b2<    >a2g2@6<        T140v12>f2a2<    c2.r4:|
o4v9q7l16 |:   v10@16e2d2          <e2>>b2@6<      v12c2c2          <<e2.r4>>:|
o4v9q7l16 |:   v10@16g2g2          <<c2>e2@6>      v12f2f2          <<g2.R4>>:|
o4v9q7l16 |:   v10@16R2R2          R2<b2@6>        v12a2a2          R2.R4:|
```

この楽譜では、３重和音の部分と４重和音の部分があります。

MMLの文法について説明します。

(1)　書式

楽譜データはMMLデータ（コマンド、パラメータ）を文字列として続けて記述します。音楽演奏のさいは、記述した順番に実行されることになります。各データは、詰めて書いてもスペースが入っていてもかまいませんし、小文字で書いても大丈夫です。

MMLデータの一覧を示します。以下、この表をもとに説明をしていきます。

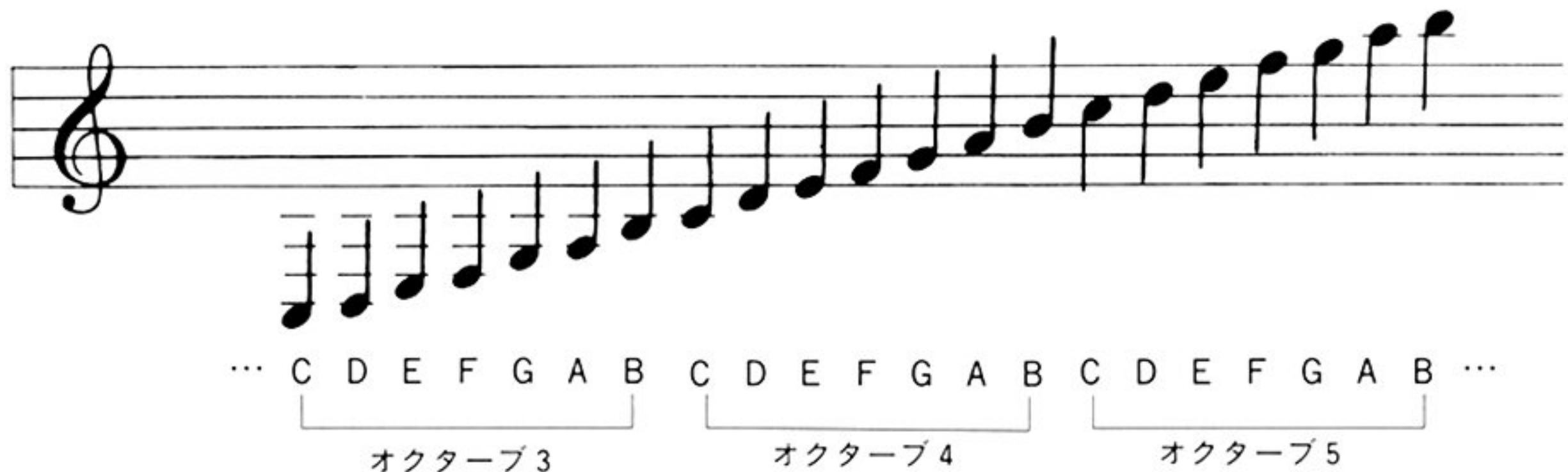
MMLデータ	意　　　味	パラメータ範囲(初期値)
An～Gn	音符 A～G：音程（後ろに#または＋を付けるとシャープ、－を付けるとフラット） n：音長（n分音符）	 1～64、省略可
Rn	休符（n：音長）	1～64、省略可
．	符点（音長を1.5倍にする）	
&	タイ（前後の音をつなげる）	
{音程データ}n	連符(n：合計の音長)	1～64
On	オクターブ(n：高さ)	0～8　(4)
＜	オクターブを1つ上げる	
＞	オクターブを1つ下げる	
Tn	テンポ(1分間に4分音符をn拍)	32～200　(120)
Ln	音長(n分音符)	1～64　(4)
@Ln	音長の細かい設定(n/192分音符)	1～192　(48)
Qn	1音中で実際に音を出す割合(n/8)	1～8　(8)
Vn	音量(n：大きさ)	0～15　(8)
@Vn	音量の細かい設定	0～127
Pn	ステレオ出力(1：左、2：右、3：左右)	1～3　(3)
@n	音色(n：音色番号)	1～200　(1)
Yr, d	OPMに直接データを書き込む r：レジスタ番号 d：データ	
@Wn	OPMの状態維持(n：音長)	1～64、省略可
@Nn	チャンネル変更	1～25
@Pn	パンポットの設定	0(右)～127(左)
@Un	ベロシティの設定	0～127
@Dn	ダンパーオンオフの設定	0(off)　1(on)
@Bn	ピッチベンドの設定 type1	0～8192～16383 (半音≒683)
@Bn1,n2	ピッチベンドの設定 type2	n1＝0～127(low byte) n2＝0～127(hi byte)
@Mn1,n2	モジュレーションの設定 n2を省略した場合は198 (speedはFM音源のみに有効)	n1＝0～127(depth) n2＝0～255(speed)

MMLデータ	意　　　　味	パラメータ範囲(初期値)
' '	和音の指定 　最初の音程のみ長さを指定可 　異なる長さの音は混在不可 最大８音(＞ ＜ ＃ － ＋ は設定可) 　(ドラムパートなどの１トラック化が可能) MIDI以外のチャンネルでは和音演奏不可 例 'C8EG<C>'	
|：n	繰り返しの始め(n：繰り返し回数)	1～256、省略値2
：|	繰り返しの終わり	
|n	繰り返しのn回目で演奏するデータの始め	1～256
〔D.C.〕	先頭に戻る(１回のみ)	
〔D.S.〕	〔SEGNO〕に戻る(１回のみ)	
〔SEGNO〕、〔$〕	〔D.S.〕からの飛び先	
〔TOCODA〕、〔*〕	〔CODA〕に飛ぶ(１回のみ)	
〔CODA〕	〔TOCODA〕からの飛び先	
〔FINE〕、〔^〕	〔D.C.〕、〔D.S.〕、〔*〕実行後の演奏終了	

⑵　音符

楽譜データとしてもっとも基本になるのは音符の記述です。実際の楽譜を見るとわかるように、ひとつの音符は音程と音長からなっています。MMLの場合も同様に、A～Gで音程を、1～64の数字で音長を表し、2つ合わせてひとつの音符になります。A～Gの代わりにRを使って休符を表すこともできます。また、音符の要素としてはタイや連符もあり、これらも簡単に記述できるようになっています。

●音程

音程データA～Gの意味を下図に示します。ただし、ここに示した音の高さは実際には音色によってオクターブが異なることがあります。

ここで、A～Gの後ろに#または+を付けると半音上、-を付けると半音下を表します。たとえば、"C#"、"C+" "D-" はすべて同じ音、音楽記号のC#（D♭）になります。

●音長

音長データは音程データ（休符を含む）の後ろに数字を付けて表し、この数字をnとするとその音符はn分音符です。たとえば、これを音楽記号で表すと次のようになります。

音長データは省略可能で、その場合は後述のLコマンドで設定された音長になります。また、ピリオド（.）を符点の意味で使用することができ、これを付けると音長が1.5倍になります。2つ付けると1.75倍です。たとえば、"C4." は符点4分音符、L8に設定されているときの "C." は符点8分音符となります。

●タイと連符

2つの音符データの間に&を入れることにより、2つの音をつないで鳴らす（タイ）ことができます。

連符は音程データを{　}でくくり、その後ろに音長データを付けることにより記述します。この場合、{　}内の音を合計すると指定した音長になるように演奏されます。つまり、1つのひとつの音は、指定した音長を音程データの個数で等分した長さで鳴らされるわけです。たとえば、これを使って3連符を記述した場合の音楽記号との関係は次のようになります。

(3)　状態設定

このMMLは音符を記述するだけでなく、オクターブやテンポ、音長、音量などを細かくコントロールすることが可能です。さらに、OPMの特長を生かして、音色の選択やステレオ出力の切り換えも行うことができます。

●オクターブ

オクターブはOコマンドで値を直接指定する方法と"<"、">"でオクターブを上げたり下げたりする方法があります。前者は音楽の各パートの先頭部分、あるいはパートの途中で音色を変えたときなど、後者は演奏中に一時的にオクターブが変わるときに使用するとよいでしょう。たとえば、次の楽譜は"GAB<CDC>BAG"のように記述できます。

オクターブの値と音程の関係は音符の項で述べたとおりです。オクターブ0ではC、C#、Dの音は使えず、オクターブ8ではC、C#、D以外の音は使えません。オクターブの値は初期状態では4になっています。

●テンポ

テンポの設定はTコマンドで行います。つまり、曲中で演奏のテンポを変えることもできるわけです。設定値は1分間に4分音符を何拍子打つかで表します。

●音長

音長は音符データのほうで1音1音について記述することができますが、これを省略した場合の音符の音長を設定するのがLコマンド、@Lコマンドです。楽譜のなかでもっとも多く出てくる音長を設定しておくことによって、効率的に楽譜データを記述することができます。

Lコマンドの設定値は音符の音長データとまったく同じ意味です。これに対して、@Lコマンドは全音符（4分音符4つ分の長さ）を192として音の長さを指定するので、音長をより細かくコントロールすることができます。たとえば、L 8と@L24は同じ意味で、@L25はL 8よりも少しだけ長い音になります。@Lコマンドによる微妙な音長設定を曲全体に対して行うと、リズムの管理が大変になるので、曲のなかで部分的に特殊な効果を出したい場合、あるいは効果音などに使用するとよいでしょう。

●音を出す時間

一般に、同じ4分音符でも、そのなかで実際に音を出している時間はまちまちです。音楽記号にテヌートとかスタッカートというのがありますが、これもそういった音の出し方を指示する記号で、音符に示された長さいっぱいまで音を出したり、逆に音を切ったりすることを示します。

このように、1つひとつの音符の音長のうち実際に音を出す時間の割合を設定するのがQコマンドです。Qコマンドではその割合を1／8～8／8で指定し、値が小さくなるほどスタッカートぎみに、大きくなるほどテヌートぎみになります。ただし、音色によってはそれほど効果はありません。なお、初期状態ではQ 8になっています。

●音量

音量はVコマンド、あるいは@Vコマンドで設定します。Vコマンドの設定値は0～15、@Vコマ

ンドの設定値は0～127で、いずれも値が大きいほど音量も大きくなります。Vコマンドと@Vコマンドの設定値にはLコマンドと@Lコマンドのような相関関係はなく、Vコマンドは聴感上V0からV15まで正比例的に音量が変化するのに対して、@Vコマンドは設定値が大きくなるほど変化量が大きくなっていきます。初期状態ではV8に設定されています。

●ステレオ出力

OPMがステレオ出力をサポートしているのは前述のとおりです。本機の内蔵スピーカはモノラルで、左右両方の音を一緒に出していますが、後面のオーディオ出力端子からオーディオアンプなどに接続することによりステレオサウンドを楽しむことができます。MMLで左右に音を振り分けるにはPコマンドを使います。P1とすれば左から、P2とすれば右から、P3とすれば左右両方から音が出ます。初期状態ではP3になっています、Pコマンドは、音色と音符データの間に入れて指定します。

●音色

音色の選択は@コマンドで音色番号を指定することで行います。初期状態では1番から68番までの音色があらかじめ登録されており(これをプリセット音といいます)、さらにユーザーが作成した音色データを登録することによって合計200番までの音色を使うことができます。また、登録済みの音色データを読み出し、データを書き換えて再登録することで、音色を修正することも可能です。

以下に、プリセット音の一覧を掲載します。このなかで打楽器はそれらしい音になる音程が限られています。括弧内に基本となるオクターブを示しておきますので参考にしてください。

1	A.ピアノ	24	ボイス		47	バスドラム	[0]
2	H.ピアノ	25	コーラス		48	タムタム	[2]
3	エレクトリックピアノ	26	グラスハープ		49	ティンパニ	[2,3]
4	クラビネット	27	ホイッスル		50	ボンゴ	[2,3]
5	セレスタ	28	ピッコロ		51	ティンパレス	[3]
6	チェンバロ	29	フルート		52	トライアングル	[3,4]
7	アコースティックギター	30	オーボエ		53	カウベル	[3,4]
8	エレクトリックギター	31	クラリネット		54	チューブラーベル	
9	ウッドベース	32	バスーン		55	スチールドラム	
10	エレクトリックベース	33	サックス		56	グロッケン	
11	バンジョー	34	トランペット		57	ビブラフォン	
12	シタール	35	ホルン		58	マリンバ	
13	ハープ	36	トロンボーン		59	クローズハイハット	[3]
14	琴	37	チューバ		60	オープンハイハット	[4]
15	パイプオルガン1	38	ブラス1		61	シンバル	[4]
16	パイプオルガン2	39	ブラス2		62	シンセサイザ1	
17	エレクトリックオルガン	40	ハーモニカ		63	シンセサイザ2	
18	アコーデオン	41	オカリナ		64	アンビューランス(効果音)	
19	バイオリン	42	リコーダー		65	嵐(効果音)	
20	チェロ	43	サンバホイッスル	[3,4]	66	レーザーガン(効果音)	
21	ストリングス1	44	パンフルート		67	ゲーム1(効果音)	
22	ストリングス2	45	スネアドラム	[2]	68	ゲーム(効果音)	
23	ピチカート	46	リムショット	[3]			

● OPM制御

これまでに説明したコマンドだけでも非常にきめ細かな音楽制御が可能ですが、さらにOPMに直接データを書き込んでコントロールすることもできます。これを行うのがYコマンドです。書式は"Yr,d"で、rにOPMのレジスタ番号、dに書き込むデータを指定します。ただし、レジスタ番号1、16、17、18、20は使えません。

Yコマンドで設定したOPMの状態を一定時間そのまま維持するのが@Wコマンドです。設定値は音符データの音長と同じ意味で、省略も可能です。このコマンドはYコマンドでOPMにKEY ON／OFFを指定したあとでのみ有効です。

下表にレジスタ番号とデータの対応を示します。各データはビットごとの意味をもっています。内容については、次項の「FM音源の音色データ（レジスタ）」を参照してください。

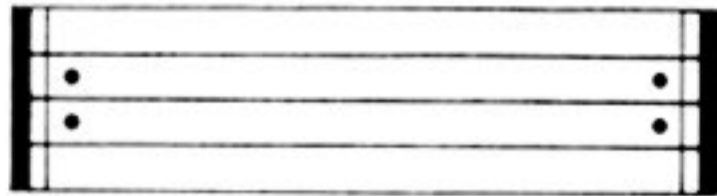
レジスタ番号（10進数）	データ（ビットデータ）							
	7	6	5	4	3	2	1	0
01	TEST（テスト用）						LFO RESET	
08		KEY ON/OFF						
		スロットマスク				チャンネル番号注1		
0F	NE			NOISE FREQUENCY				
10	CLK A1							
11							CLK A2	
12	CLK B							
14	CSM		F RESET		IRQEN		LOAD	
			B	A	B	A	B	A
18	スピード（LFRQ）							
19	PM/AM	PMD or AMD						
1B	CT2（未使用）	CT1（未使用）					ウェーブフォーム	
20～注2	L,R PAN		フィードバック			アルゴリズム		
28～注2		KEY CODE						
		OCT			NOTE			
30～注2	KEY FRACTION							
38～注2		PMS					AMS	
40～注3		DT1			MUL			
60～注3		TL						
80～注3	KS			AR				
A0～注3	AMS-EN			D1R				
C0～注3	DT2			D2R				
E0～注3	D1L				RR			

注1：チャンネル番号＝0～7
注2：レジスタ番号＝表中のレジスタ番号＋チャンネル番号
注3：レジスタ番号＝表中のレジスタ番号＋（オペレータ番号－1）×4＋チャンネル番号

(4)　繰り返し

この MML では繰り返しがサポートされ、楽譜データを効率的に記述することができます。また、これを活用することによってトラックバッファに書き込まれるデータを大幅に節約することができますので、長い曲を演奏するときにも有利です。繰り返しを示すコマンドは一般の音楽記号とよく似たものになっていますから、音楽記号と対応させて説明していきます。

●単純ループ

音楽記号に次のようなものがあります。

これは 𝄆 と 𝄇 の間を繰り返すものです。MML でも同様に繰り返したい内容を｜：と：｜でくくることによって繰り返しをすることができます。ここで｜：の後ろに 1 ～256の数字を付けることで繰り返し回数を指定することが可能です。省略した場合は 2 回になります。｜：～：｜ループを含む楽譜データをさらに｜：～：｜でくくることはできません。

また、音楽記号には⌈1.　とか⌈2.　のようなものもあります。これは繰り返しの何回目かに演奏する内容を指定します。これも MML でサポートされており、｜n で n 回目に演奏する内容を指定することができます。たとえば、

のような楽譜は ”｜：CDEF｜1GFED：｜｜2GAB<C” と記述します。

●始めに戻る

音楽記号で D.C.というのは「始めに戻る」という意味です。MML でも同様に[D.C.]と中括弧でくくって記述することで曲の始めに戻ることができます。

● Segno に戻る

音楽記号の𝄋はセーニョ(Segno)と読みます。そして、D.S.というのはダルセーニョと読み「Segno に戻る」という意味です。それぞれ[SEGNO]、[D.S.]と記述すれば MML でも同じことをします。[SEGNO]は[$]と略記することが可能です。

● Coda に飛ぶ

音楽記号に𝄌というのがあります。これはコーダ (Coda) と読み、D.C.や D.S.を実行したあと to 𝄌 (to Coda) に来たときにそこへ (Coda) 飛ぶためのものです。MML ではそれぞれ[CODA]、[TOCODA]と記述します。後者は[＊]と略記することが可能です。

● Fine

D.C.や D.S.で繰り返しをしたあとで曲が終了する場合に、それを示すのが Fine です。同様に、MML では、[FINE]あるいは[^]を実行すると演奏が終わります。

●OPMコマンド

FM音源やMIDIを直接制御する場合に使用するのが、OPMコマンドです。

以下のOPMコマンドは､OS上からデバイス名'OPM' または'MIDI' に直接出力して実行することができます。

具体的には、スクリーンエディタEDなどを使って、MMLと共にアスキーファイルとして作成し、コマンドモードから

COPY 〈ファイル名〉 OPM ↵

COPY 〈ファイル名〉 MIDI ↵

と入力すると実行することができます。

OPMコマンド	パラメータ ／ 機　　能
(REM) または／	"(REM)"または"／"で始まる行を注釈行とみなします (全角文字で始まる行も注釈とみなします)
(／OPM) (／OMP) (／MOP) (／MPO) (／PMO) (／POM)	OPM＝ch1～8　PCM＝ch9　MIDI＝ch10～25 OPM＝ch1～8　MIDI＝ch9～24　PCM＝ch25 MIDI＝ch1～16　OPM＝ch17～24　PCM＝ch25 MIDI＝ch1～16　PCM＝ch17　OPM＝ch18～25 PCM＝ch1　MIDI＝ch2～17　OPM＝ch18～25 PCM＝ch1　OPM＝ch2～9　MIDI＝ch10～25 (ch1～25はチャンネル1～チャンネル25を示します) 各音源に対するチャンネル番号の割り当てを変更します ドライバ組み込み時のオプションスイッチ（"／OPM","／POM"など）と、同じ機能です
(／Yn)	MUSIC PRO-68K［MIDI］で作成された'～.MUS' ファイルを演奏する場合は、(／Y0) を指定してください。その他の場合は、(／Y1) を指定してください デフォルトは（／Y1）に設定されています
(／Pn, filename)	n＝ノート番号 (0～127) filenameで指定されたPCMデータのファイルを読み込み､n番のノート番号に登録します。ファイル名はフルパス名で指定してください ファイル名はそのままPCMの音色名として登録されます
(／An, tonename)	n＝ノート番号 (0～127) n番のノート番号にPCMの音色名を登録します

OPMコマンド	パラメータ ／ 機　　能
(／S,filename)	SOUND　PRO-68Kの '～.SND' ファイル（または (／SV,filename,n) で作成されたファイル）を読み込み、FM音源に登録します ファイル名はフルパス名で指定してください 拡張子'.SND' は不要です。FM音源の音色名も同時に登録します
(／SV, filename, n1,n2. .n3,n4-n5)	n＝音色番号（1～200) SOUND PRO-68Kの'～.SND'ファイルのフォーマットで現在登録されている音色データの内、n で指定された音色データをファイルに保存します ファイル名はフルパス名で指定してください ファイル名に拡張子 '.SND' は不要です n を省略した場合は、200音色すべてファイルに登録します 　　例)　　(／SV,TEST1,1-8,20-30,180-) 音色番号1～8,20～30,180～200の音色データをTEST1.SNDのファイル名でファイルに保存します
(／Vn,tonename)	n＝1～200 FM音源の音色名を登録します
(／SPn)	n＝0～255（デフォルトは198) FM音源のLFO（モジュレーション）の変化する速さを設定します (I n) でデフォルトに設定されます
(／En)	n＝0 off　n＝1 on（デフォルトは1） FM音源に設定する音色パラメータの内、LFQ／PMD／AMDなどの全体に影響するパラメータを音色切り替え時にセットするかどうかを選択します すべてのチャンネルに対してモジュレーションを正常に機能させるためには、nに 0 を指定しておいてください
(／n)	n＝0～127（デフォルトは127) マスターボリュームを設定します。(I n) を実行するまで有効です PCM音に対しては無効です
(／PI)	PCMバッファをクリアします
(I n)	n＝0 or 1 トラックデータを初期化します。(M t,s) も初期化されます 各トラックバッファは 4 バイトに設定されます n＝0または省略した場合は、音色データを初期化しません n＝1で音色データも初期化します
(V n1,n2,d1,d2. ･･dm･･d55-n2)	n1＝音色番号（1～200) n2＝書き換えを始めるパラメータ番号（0～54) dm＝音色パラメータ（0～255) FM音源の音色を設定します n2は、パラメータの何番から書き換えるかを示しています

OPMコマンド	パラメータ　／　機　　能
(D n1,n2)	n1＝コピー元の音色番号（1～200） n2＝コピー先の音色番号（1～200） FM音源のn1番の音色をn2番にコピーします
(M t,s)	t=トラック番号　s=バッファサイズ トラックバッファを確保します tを省略すると、すべてのトラックにsを設定します (I）でデフォルト（全トラック4バイト）に設定されます 確保したバッファ容量が不足していると、正常に演奏されない場合があります (M,T,S)コマンドでエラーが起きた場合は、その時点での確保可能な最大値を設定します
(A c,t)	c=チャンネル番号　t=トラック番号 チャンネルcで演奏するトラック番号を指定します cが重複した場合は、後から指定したチャンネルに設定されます tが重複した場合は、複数のチャンネルで同じトラックが演奏されます
(O n)	n=テンポデータ（20～300） テンポを設定します
(Q n1,n2)	n1＝分子　n2＝分母 拍子を設定します
(T n)MMLdata..	n=トラック番号（1～80） 指定のトラックにMMLデータを書き込みます
(E n1,n2,..n25)	n=チャンネル番号（1～25） nで指定されたチャンネルの音声出力を許可します。nで指定されたチャンネル以外のチャンネルは音声出力が禁止されています (内部的な演奏は継続しています) nがすべて省略された場合は、すべてのチャンネルの音声出力が許可されます
(P n1,n2,..n25)	n=チャンネル番号 nで指定されたチャンネルの演奏を開始します nがすべて省略された場合、すべてのチャンネルの演奏を開始します MIDIのチャンネルに対しては、スタートコード（$FA）を出力します
(S n1,n2,..n25)	n=チャンネル番号 nで指定されたチャンネルの演奏を停止します nがすべて省略された場合、すべてのチャンネルの演奏を停止します MIDIのチャンネルに対しては、ストップコード（$FC）を出力します

<table>
<tr><th>OPMコマンド</th><th>パラメータ ／ 機 能</th></tr>
<tr><td>(C n1,n2,..n25)</td><td>n=チャンネル番号
nで指定されたチャンネルの演奏を再開します
nがすべて省略された場合、すべてのチャンネルの演奏を再開します
MIDIチャンネルに対しては、コンティニューコード ($FB) を出力します
常にMIDIクロック ($F8) を出力しているので、外部シーケンサーと同期演奏させることができます</td></tr>
<tr><td>(W n1,n2,..n25)</td><td>n=チャンネル番号
nで指定されたチャンネルが終了するまで待ちます
(ESCキーまたはBREAKキーで中止します)
nがすべて省略された場合、すべてのチャンネルの演奏が終了するまで待ちます</td></tr>
<tr><td>(PM n)</td><td>n=小節番号
演奏を開始する小節番号を指定します
(S n) または演奏終了で解除されます
(Q n1,n2) で曲の拍子が設定されていなければ正常な演奏ができません
また、全トラックにおいて、MMLデータ、"|:"、":|"、"[D.S.]"、"[D.C.]"、"[$]"などの数が同じでなければなりません</td></tr>
<tr><td>(SM n)</td><td>n=小節番号
演奏を停止する小節番号を指定します
(S n) または演奏終了で解除されます
(Q n1,n2) で曲の拍子が設定されていなければ正常な演奏ができません
また、全トラックにおいて、MMLデータ、"|:"、":|"、"[D.S.]"、"[D.C.]"、"[$]"などの数が同じでなければなりません</td></tr>
<tr><td>(N c,n)</td><td>c=チャンネル番号 (1～25) n=出力チャンネル番号 (1～25)
一時的に出力チャンネルを変更します
(I n) で元に戻ります</td></tr>
<tr><td>(G c,n)</td><td>c=チャンネル番号 n=音色番号 (OPM：1～200 MIDI：1～128)
指定のチャンネルに音色番号を設定します</td></tr>
<tr><td>(L c,n)</td><td>c=チャンネル番号 n=ボリュームデータ (0～127)
指定のチャンネルにボリュームデータを設定します</td></tr>
<tr><td>(B c,n)</td><td>c=チャンネル番号 n=パンポットデータ (0～127)
指定のチャンネルにパンポットデータを設定します</td></tr>
<tr><td>(U c,n)</td><td>c=チャンネル番号 n=ベロシティデータ (0～127)
指定のチャンネルにベロシティデータを設定します</td></tr>
<tr><td>(F c,n)</td><td>c=チャンネル番号 n=トランスポーズデータ (0～48)
指定のチャンネルにトランスポーズデータを設定します
±2オクターブの範囲で半音ずつ音程をシフトします (デフォルトは24)</td></tr>
<tr><td>(Y c,n1,n2)</td><td>c=チャンネル番号 n1=コントロールコード n2=データ
指定のチャンネルにコントロールデータを出力します</td></tr>
</table>

OPMコマンド	パラメータ　／　機　　　能
(J c)	c=チャンネル番号（1～25） 指定のチャンネルへオールノートオフを出力します cを省略した場合は、すべてのチャンネルにオールノートオフを出力します
(X n1,n2,... ...n512)	n1～nmまでのデータをMIDIへ直接出力します 指定できるデータの数は512個までです
(Z n)	n=0 内部同期　n=1 外部同期　n=2 FSK同期

ノート番号：音程　対応表

ADPCMで取り込んだ音のデータファイルを音程に割り付け、曲の途中でその音程を演奏することで取り込んだ音を演奏中に再生することができます。
取り込んだ音データの音程への割り付けは、／Pnコマンドで指定します。
／Pnコマンドで指定するノート番号と音程の対応表は次の通りです。

	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
O 0	12	13	14	15	16	17	18	19	20	21	22	23
O 1	24	25	26	27	28	29	30	31	32	33	34	35
O 2	36	37	38	39	40	41	42	43	44	45	46	47
O 3	48	49	50	51	52	53	54	55	56	57	58	59
O 4	60	61	62	63	64	65	66	67	68	69	70	71
O 5	72	73	74	75	76	77	78	79	80	81	82	83
O 6	84	85	86	87	88	89	90	91	92	93	94	95
O 7	96	97	98	99	100	101	102	103	104	105	106	107
O 8	108	109	110	111	112	113	114	115	116	117	118	119

用例

```
(i1)
(m1,980)
(a1,1)
(／PI)
(／PI2,b：¥sample¥sample.pcm)
(t1) T100 V15 @15 O4C8E8G8<C8 @N9O0C8 @N1O5C8>G8E8C8
(P)
```

※MMLデータとOPMコマンドの具体的な使い方については、SX-WINDOWアプリケーションディスクのSAMPLEという名のディレクトリ内にあるX68K.OPMの内容を参照してください。

7.4.2　FM音源の音色データ(レジスタ)

本機のFM音源には、前述の通りYM2151 (OPM) を使用しています。これは、8オクターブステレオ出力が可能なFM　Generatorを8チャンネル持っており、その各チャンネルに、オペレータとしてモジュレータ2個 (M1, M2)、キャリア2個 (C1, C2) が対応しています。しかも、このオペレータの組み合わせや、チャンネルごとのPhase Generatorとオペレータごとのエンベロープ Generatorの設定値を変化させることにより、いろいろな音作りが可能になっています。

また、このFM音源では、低周波発振器 (LFO) をもっており、それを使用することでビブラートやトレモロ効果が生み出せます。

●FM音源の音色データ

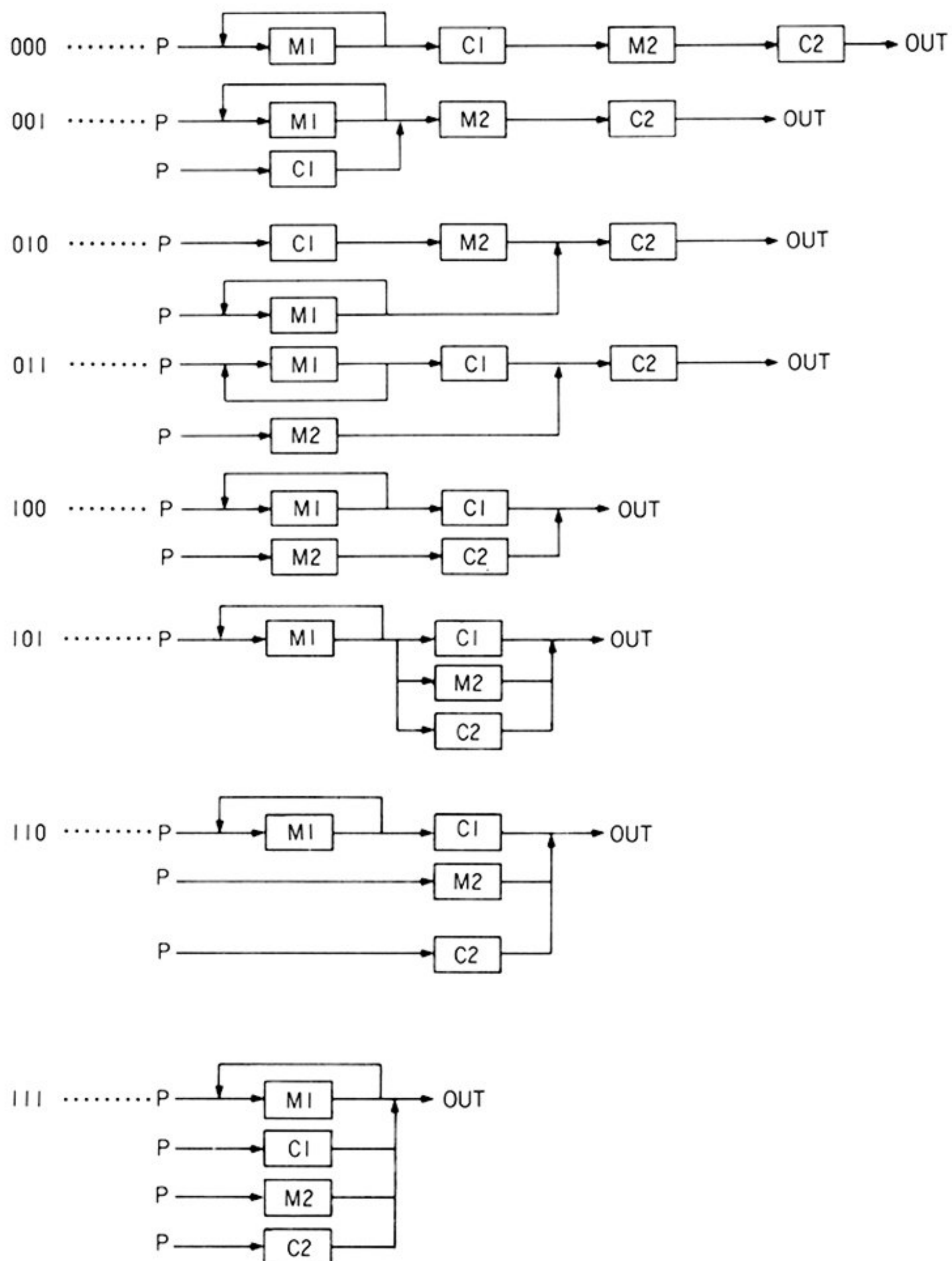
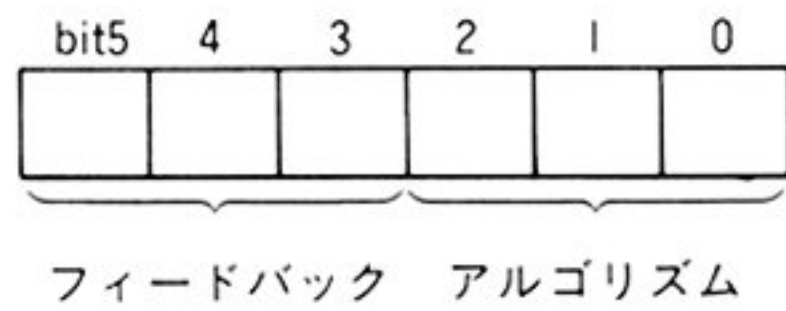
FM音源データは、次のような配列形式になっています。

パラメータ	内　　　容	設定可能範囲
0	フィードバック／アルゴリズム	(0～6)
1	スロットマスク	(0～15)
2	ウェーブフォーム	(0～3)
3	シンクロ	(0, 1)
4	スピード	(0～255)
5	PMD	(0～127)
6	AMD	(0～127)
7	PMS	(0～7)
8	AMS	(0～3)
9	L.RPAN	(0～3)
10		
11	M1／AR	(0～31)
12	M1／D1R	(0～31)
13	M1／D2R	(0～31)
14	M1／RR	(0～15)
15	M1／D1L	(0～15)
16	M1／TL	(0～127)
17	M1／KS	(0～3)
18	M1／MUL	(0～15)
19	M1／DT1	(0～7)
20	M1／DT2	(0～3)
21	M1／AMSイネーブル	(0～3)

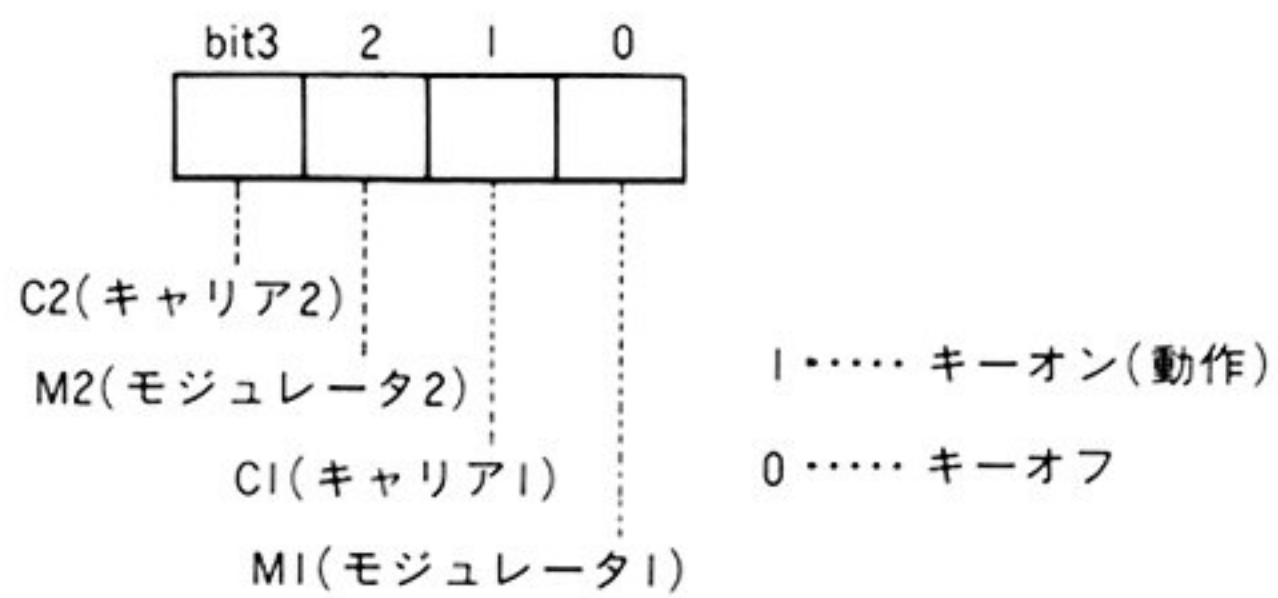
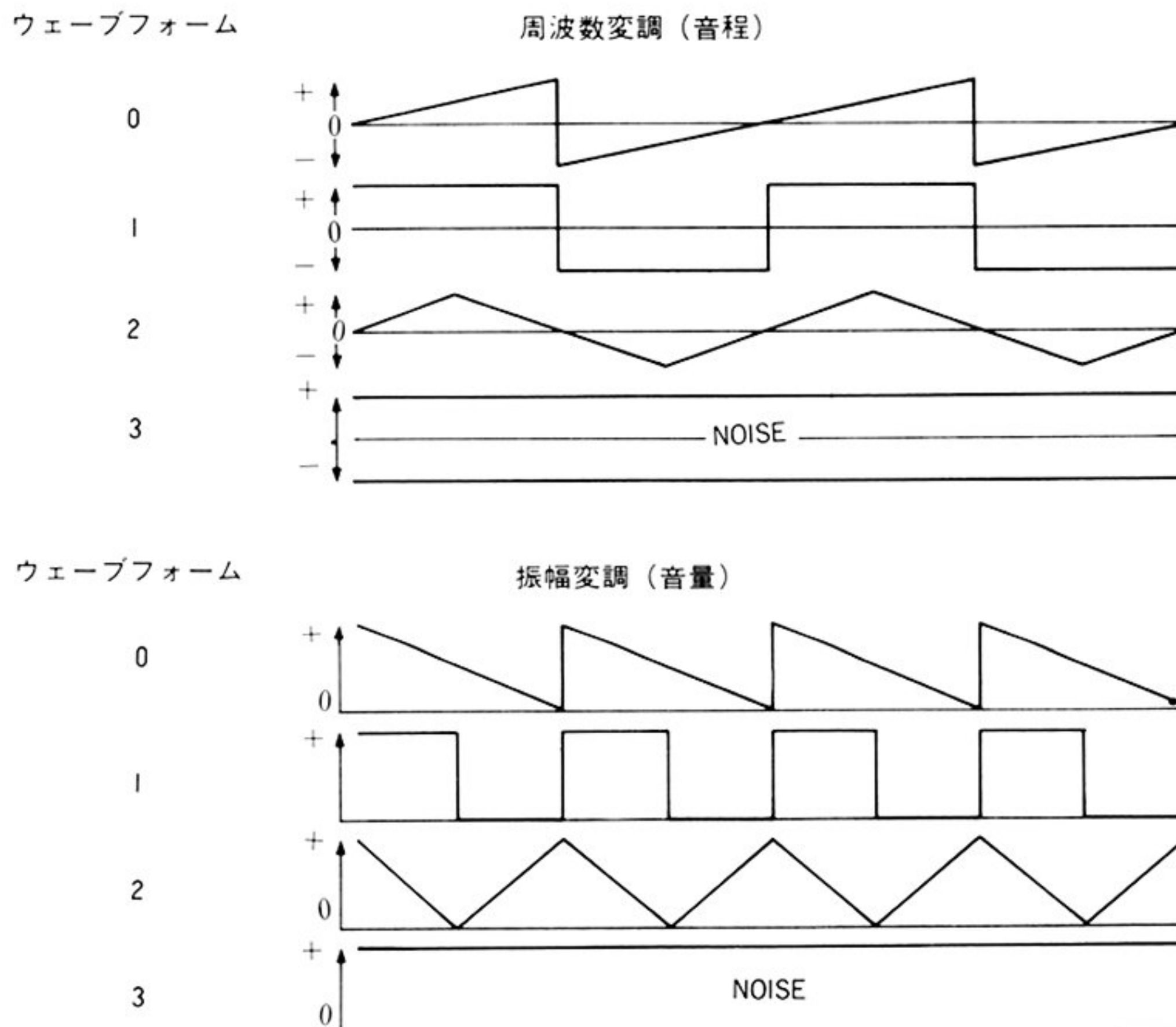
パラメータ	内　　　　容	設定可能範囲
22	C1／AR	(0～31)
23	C1／D1R	(0～31)
24	C1／D2R	(0～31)
25	C1／RR	(0～15)
26	C1／D1L	(0～15)
27	C1／TL	(0～127)
28	C1／KS	(0～3)
29	C1／MUL	(0～15)
30	C1／DT1	(0～7)
31	C1／DT2	(0～3)
32	C1／AMSイネーブル	(0～3)
33	M2／AR	(0～31)
34	M2／D1R	(0～31)
35	M2／D2R	(0～31)
36	M2／RR	(0～15)
37	M2／D1L	(0～15)
38	M2／TL	(0～127)
39	M2／KS	(0～3)
40	M2／MUL	(0～15)
41	M2／DT1	(0～7)
42	M2／DT2	(0～3)
43	M2／AMSイネーブル	(0～3)
44	C2／AR	(0～31)
45	C2／D1R	(0～31)
46	C2／D2R	(0～31)
47	C2／RR	(0～15)
48	C2／D1L	(0～15)
49	C2／TL	(0～127)
50	C2／KS	(0～3)
51	C2／MUL	(0～15)
52	C2／DT1	(0～7)
53	C2／DT2	(0～3)
54	C2／AMSイネーブル	(0～3)

(1) フィードバック／アルゴリズム

●アルゴリズム……各チャンネルごとの4つのオペレータの組み合せを設定します。8種類の組み合わせがあり、それぞれで変化に富んだ音色作りができますが、一般に、キャリアの数が少なく、モジュレータの数が多いほど複雑な音色変化を作れます。

```
000 ……… P  M1  C1  M2  C2  OUT

001 ……… P  M1  M2  C2  OUT
         P  C1

010 ……… P  C1  M2  C2  OUT
         P  M1

011 ……… P  M1  C1  C2  OUT
         P  M2

100 ……… P  M1  C1  OUT
         P  M2  C2

101 ……… P  M1  C1  OUT
                M2
                C2

110 ……… P  M1  C1  OUT
         P      M2
         P      C2

111 ……… P  M1  OUT
         P  C1
         P  M2
         P  C2
```

●フィードバック……各チャンネルごとの最初のオペレータの出力をそのオペレータ自身にフィードバックすることにより、ひとつのオペレータだけで変調が行われます。数値が大きくなるほど、大きく変調され、倍音成分が増え、音色が大きく変わります

0 0 0……　OFF
0 0 1……　$\pi/16$
0 1 0……　$\pi/8$
0 1 1……　$\pi/4$
1 0 0……　$\pi/2$
1 0 1……　π
1 1 0……　2π
1 1 1……　4π

(2)　スロットマスク……各チャンネルごとの各オペレータのキーオン（出力されます）／オフ（出力されません）を設定します。

(3) ウェーブフォーム*……LFOから発生される波形を設定します。波形の違いにより、音量や音色の変化のしかたが変わります。

(4) シンクロ*……LFOのスタートとキーオン（音が発生される瞬間）をシンクロさせます。シンクロ状態にすると音が発生されるたびにビブラートやトレモロ効果などが波形の先頭からかかります。

0……シンクロ解除

1……シンクロ状態

(5) スピード*……LFOの発振周波数（ビブラートやトレモロ効果の速さ）を設定します。数値が大きくなるほど、周波数が高くなり、スピードが速くなります。

0～255（約0.009～59Hzに対応）

(6) PMD（Pitch Modulation Depth）*

……周波数変調（音程）に対してかけるLFOの出力レベルを設定します。数値が大きくなるほど、深くかかります。

0～127（出力レベル小→出力レベル大）

(7) AMD（Amplitude Modulation Depth）*

……振幅変調（音量）に対してかけるLFOの出力レベルを設定します。数値が大きくなるほど、深くかかります。

0～127（出力レベル小→出力レベル大）

(8)　PMS (Pitch Modulation Sensitivity)*

……各チャンネルごとの周波数変調（音程）に対してかけるLFOの感度を設定します。数値が大きくなるほど、感度は大きくなります。

0～7　(感度小→感度大)

(9)　AMS (Amplitude Modulation Sensitivity)*

……各チャンネルごとの振幅変調（音量）に対してかけるLFOの感度を設定します。数値が大きくなるほど、感度は大きくなります。

0～3　(感度小→感度大)

(10)　L,R PAN……各チャンネルごとの出力方式を設定します。

0……出力禁止

1……左出力

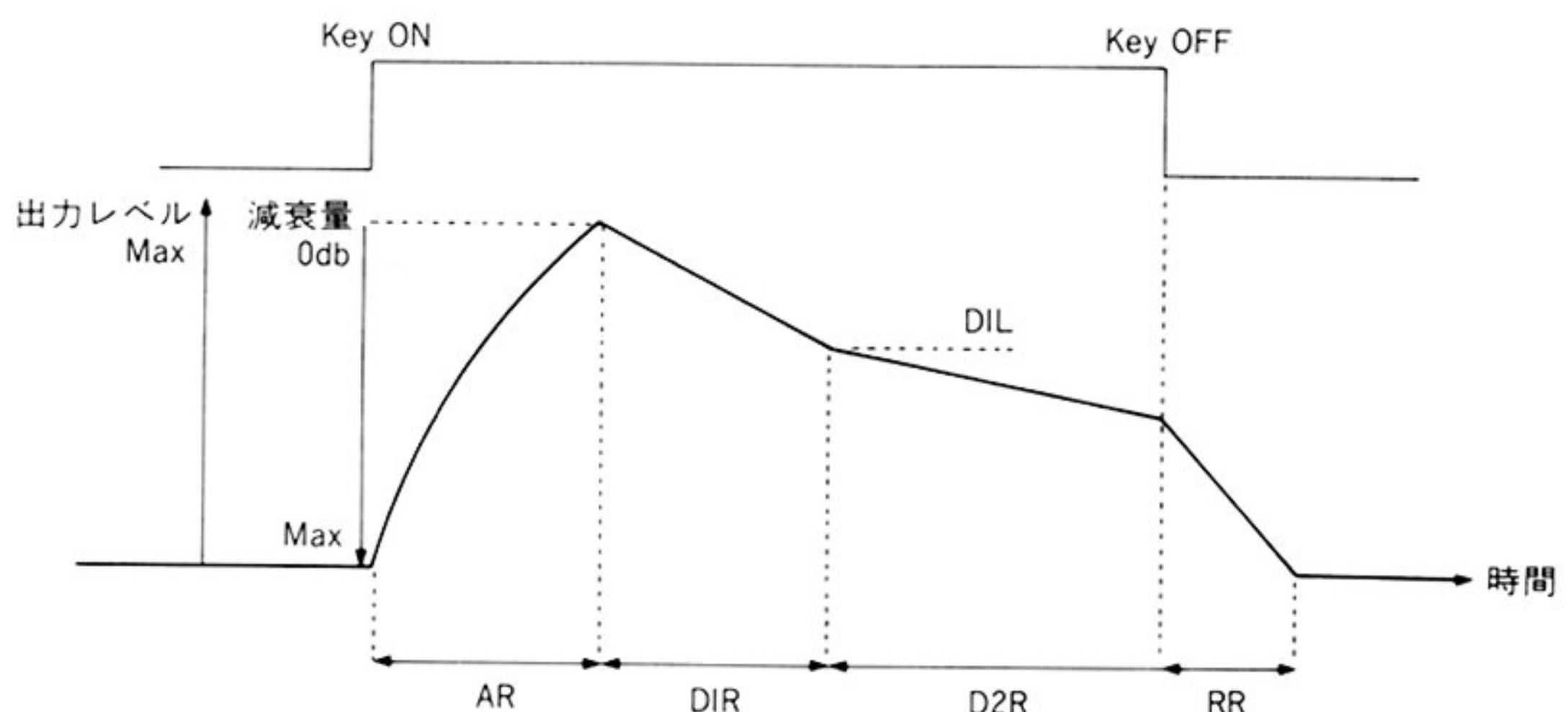
2……右出力

3……ステレオ出力

*……LFO (Low Frequency Oscillator) に関するデータ

なお、PMDが0になっていると周波数変調(音程)に対してのLFOはかかりませんし、AMDが0になっていると振幅変調（音量）に対してのLFOはかかりません。

なお、LFOを使用すると音色が変わります。

〔オペレータ単位〕…以降の説明においてオペレータ1はM1、オペレータ2はC1、オペレータ3はM2、オペレータ4はC2に対応します。

(11)　AR (Attack Rate)** ……キーオンされたあと、出力レベルが最大になるまでの速度（減衰量が0dbになるまでのレート）を設定します。数値が大きくなるほど音の立ち上がりが速くなります。

0～31

⑿ D1R (First Decay Rate)** ……出力レベルが最大になってから、D1Lで設定したレベルに下がるまでの速度（減衰量が0dbからD1Lになるまでのレート）を設定します。数値が大きくなるほど、音の減衰速度が速くなります。

0～31

⒀ D2R (Second Decay Rate)** ……D1Lで設定したレベルになってから、キーオフされるまでの速度（減衰量がD1Lになってからキーオフされるまでのレート）を設定します。数値が大きくなるほど、音の減衰速度が速くなります。

0～31

⒁ RR (Release Rate)**……キーオフされてから音が消える（最大減衰量(96db)になる）までの速度を設定します。数値が大きくなるほど、キーオフしてからの音の減衰速度が速くなります。

0～15

⒂ D1L (First Decay Level)**……D1RからD2Rに移るところの出力レベルを設定します。数値が大きくなるほど出力レベルが低くなります。

0……減衰量 0 db　　8……減衰量24db
1……減衰量 3 db　　9……減衰量27db
2……減衰量 6 db　　10……減衰量30db
3……減衰量 9 db　　11……減衰量33db
4……減衰量12db　　12……減衰量36db
5……減衰量15db　　13……減衰量39db
6……減衰量18db　　14……減衰量42db
7……減衰量21db　　15……減衰量93db

⒃ TL (Total Level)** ……音色及び音量を制御するための各オペレータにおけるEGのトータル出力レベルを設定します。キャリアの出力レベルは音量を変化させ、モジュレータの出力レベルは音色を変化させます。数値が大きくなるほど、出力レベルは小さくなります。

0～127（出力レベル大→出力レベル小）

⒄ KS (Key Scaling)** ……音の高さ（キーコード）によって、エンベロープの長さを変化させます。数値が大きくなるほど、高音域でのエンベロープの長さが短くなります（音色が鋭くなります)。

0～3

⒅ MUL (Phase Multiply)　…各オペレータの周波数を設定します。設定値は、鍵盤標準ピッチに対する周波数の比を表わします（0の場合は1/2の周波数)。

0～15

⒆ DT1 (Detune 1)　……各オペレータのピッチを微妙にずらすことにより、音の波の干渉作用を起こし、音に広がりを与えます。数値が0か4のとき、ピッチ変化は0です。数値が1、2、3の場合、ピッチ変化は＋1、＋2、＋3となり、数値が5、6、7の場合、ピッチ変化は－1、－2、－3となります。

0～7

⒇ DT2 (Detune 2)　……各オペレータの周波数を設定しますが、周波数比を非整数倍にしたいときに設定します。数値が0のとき、MULで設定された周波数を1.00倍、1のとき1.41倍、2のとき1.57倍、3のとき1.73倍に変えます。

0～3

(21) AMSイネーブル………各オペレータごとにAMSのイネーブルを設定します。

0……AMSディセーブル

1……AMSイネーブル（AMS動作）

＊＊……EG (Envelope Generator) に関するデータ

キャリアのオペレータのエンベロープ・ジェネレータは音量の時間的変化を、モジュレータのオペレータのエンベロープ・ジェネレータは音色の時間的変化をコントロールします。

7．5　プリンタ設定について

プリンタを使用するためには、CONFIG. SYSファイル中で、DEVICE行にプリンタドライバを指定する必要があります。

次に使用するプリンタと、そのためのプリンタドライバとの対応表を示します。

プリンタドライバ	使用するプリンタ
PRNDRV. SYS	CZ8PK3～10, CZ8PN1, CZ8PC1～5, CZ8PG1～2
PRNDRV1. SYS	ESC/P24-J81レベル以上のプリンタ (VP-85K、VP-135K(以上エプソン)など)
PRNDRV2. SYS	PC-PR201(NEC)
PRNDRV3. SYS	NM-9300(NEC), AR-2400(スター)

お買い上げいただいた時点では、プリンタドライバとしてPRNDRV. SYSが設定されていますので、そのまま、表に示した16機種のプリンタを使用することができます。これら16機種以外で、表中に掲げられたプリンタをご使用の際は、次のようにしてCONFIG. SYSファイルを変更してください。

1) 本書第1部「2．2　Human68kの起動」の手順にしたがって、Human68kのコマンドモードにします。
2) 次のように入力して、スクリーンエディタEDを実行します。

```
ED CONFIG.SYS ↵
```

画面にCONFIG.SYSファイルの内容が表示されますので、第2部のスクリーンエディタEDを参照して変更してください。

たとえば使用するプリンタがPC-PR201(NEC)であれば、デバイスの中のプリンタを¥SYS¥PRNDRV.SYSから、次のように変更します。

```
¥SYS¥PRNDRV2.SYS
```

3) 設定し直したら、ESC・E（ESCを押し、続けてEを押す）でスクリーンエディタEDを終了してください。
4) リセットスイッチを押して、システムを再起動します。これで、新しく設定したプリンタを使用することが可能となります。

● PRNDRV. SYSデバイスドライバの機能について

PRNDRV. SYSデバイスドライバについては、CONFIG. SYSファイル中のDEVICE行で、行末に次のようなパラメータを付加することにより、プリンタ印字の方法を各種に設定することができます。

```
DEVICE=¥SYS¥PRNDRV.SYS  #/Mn
```

ここで、nには次の値のうちのいずれかを指定します。

n	印字方法
0	すべての文字がコード印字
1	英数字、カナ文字、JIS第1・第2水準漢字はコード印字、外字のみビットイメージ印字
7	英数字、カナ文字、JIS第1水準漢字はコード印字、第2水準漢字、外字はビットイメージ印字
15	英数字、カナ文字はコード印字、JIS第1・第2水準漢字、外字はビットイメージ印字
31	すべての文字がビットイメージ印字

なお、#／Mnパラメータを指定しない場合は、#／M7を指定したのと同様となります。

たとえば、CZ-8PK6は、JIS第1・第2水準漢字が内蔵されていますので、これらをコード印字にしたい場合、次のようにします。

```
DEVICE=¥SYS¥PRNDRV.SYS  #／M1
```

7．6　RAMディスクについて

本章のDEVICEコマンドで説明したとおり、本機では、グラフィックRAM、メインメモリ、スタティックRAMを、RAMディスクとして使用することができます。登録を行うには、スクリーンエディタEDやSX-WINDOWの日本語マルチフォントエディタを使ってCONFIG.SYSファイル中に次のような行を追加した後、リセットを行ってください。

(1)　グラフィックRAMをRAMディスクとして使用する場合

```
DEVICE=¥SYS¥RAMDISK.SYS  #G
```

この行を登録することにより、グラフィックRAM512Kバイト分をRAMディスクとして使用することができます。

なお、他のアプリケーションソフト（たとえばX-BASICなど）でグラフィックRAMを使用すると、RAMディスクの内容は壊されますので注意してください。

(2)　メインメモリをRAMディスクとして使用する場合

```
DEVICE=¥SYS¥RAMDISK.SYS  #Mnnn
```

ここで、nnnには、RAMディスクとして使用する容量をKバイト単位で指定します。(1)のグラフィックRAMとともに同時に指定することも可能です。たとえば、

```
DEVICE=¥SYS¥RAMDISK.SYS  #GM128
```

とすると、グラフィックRAM512Kバイトと、メインメモリ128Kバイトの順にRAMディスクが割り付けられます。

(3) スタティックRAMをRAMディスクとして使用する場合

DEVICE=¥SYS¥SRAMDISK.SYS

この場合は、この行をCONFIG.SYSに登録した後、Human68kのコマンドモードから

SWITCH　S=R ↵

と入力します。この後にリセットを行ってください。

逆にスタティックRAMをRAMディスクとして使用するのをやめる場合は、

SWITCH　S=N ↵

と入力します。

なお、以上のいずれの場合もRAMディスクの内容を初期化するときは、SHIFT キーを押しながらリセットスイッチを押してください。

(4) RAMディスクの利用法

RAMディスクは、CONFIG.SYSの"DEVICE="で指定した順にドライブ名が割り当てられます。たとえば、ハードディスクが1台、フロッピーディスクが2台のシステムの場合、これらのドライブにはA、B、Cというドライブ名が割り当てられ、さらにCONFIG.SYSで次のように指定していると、

DEVICE=¥SYS¥RAMDISK.SYS　#G
DEVICE=¥SYS¥SRAMDISK.SYS

グラフィックRAMのRAMディスクがドライブDに、スタティックRAMのRAMディスクがドライブEに割り当てられます。こうして割り当てられたドライブに対しては、通常のドライブと同じように扱うことができます。

●RAMDISK.SYSについて

RAMDISK.SYSでは、グラフィックRAMをRAMディスクに使用することができます。しかし、X-BASICを含め多くのアプリケーションでグラフィックRAMが使われるので、グラフィックRAMをRAMディスクとして使わない方が無難です。逆にソフトウェア開発に利用する場合、グラフィックRAMを使用することがほとんどないので、積極的に利用するとよいでしょう。

メインメモリをRAMディスクに使用する場合、設定した分だけメインメモリとして利用できるメモリ容量が減ります。ですから、あまり大きな値を設定すると、メモリ不足で動作しないアプリケーションもでてきます。その場合は、設定値を変更するか、別売の増設メモリを増設するかのどちらかの方法をとってください。

RAMディスクはハードディスクよりも高速ですから、使い方によってはたいへんに便利なものですが、本体前面の電源スイッチを切るとその内容が消えてしまうという欠点もあります。そのため、RAMディスクにはプログラムファイルなどの更新することのないファイルを格納し、データファイルなどはハードディスクあるいはフロッピーディスクに格納するようにしてください。

第8章　デバイスエラーの処置方法

8．1　イントロダクション

Human68kは、何らかのエラーが発生すると、これを検知し、エラーメッセージを画面に表示してそのことを知らせます。したがって、エラーメッセージが表示された場合は、メッセージに従って適切な処置を行ってください。

8．2　デバイスエラーメッセージ

ディスクドライブ、ディスプレイ、プリンタなどのデバイスに対して読み込みおよび書き込みを実行しているときにエラーが発生した場合、次のような形でエラーメッセージが表示されます。

〈エラーのタイプ〉
中止〈A〉　再実行〈R〉　無視〈I〉?

各項目の内容は次のとおりです。

● **〈エラーのタイプ〉**
発生したエラーの種類を表すメッセージです。どういう状況が発生したかを示すものです。

● **中止〈A〉　再実行〈R〉　無視〈I〉?**
エラー発生後の処置をどうするかの応答を求めるメッセージです。指定のキーを押すとエラーメッセージは画面から消え、処置が行われます。処置方法については、次の8．3節をご覧ください。

8．3　デバイスエラーの処置方法

デバイスエラーが表示されたときは、〝A〞、〝R〞、〝I〞のいずれかを押してください。それぞれの意味は、次のとおりです。

〝A〞〈Abort〉

読み出しまたは書き込みを中止します。ただちにHuman68kに戻ります。〝R〞で再実行してもエラーが繰り返されるときには、〝A〞を押してください。

〝R〞〈Retry〉

処理を再実行します。ふつう、デバイスエラーメッセージが表示されたときは、まずこの〝R〞を押してください。

〝I〞〈Ignore〉

エラーを無視して、そのまま処理を続行します。複数のファイルの処理などをしていて、処理を先に進めたいときには、"I"を選択します。ただし、ディスク上のデータが破壊されることもありますので注意が必要です。

第2部

スクリーンエディタED

第1章　スクリーンエディタEDの概要

この章ではスクリーンエディタ "ED" の使い方について解説します。

1.1　EDとは

EDは、ソースプログラムやテキストファイルを作成・編集するときに使用する、ツールプログラムです。EDは豊富なコマンドを備えており、CONFIG.SYS ファイルや AUTOEXEC.BAT ファイルを作成するときなどにも便利に使うことができます。EDはシステムディスク上の "BIN" というディレクトリの中にあります。

1.2　EDの機能

EDは次のような機能を備えています。

● 複数にわたって編集することができます。
● 1行の文字数は最大128、256、512、1024文字のうちから選ぶことができます。
● 編集画面の横スクロールを行うことができます。これにより、画面の端で行が折り返されることなく、長い行を入力できます。
● 一度に10個までのファイルの編集が可能です。
● 切り貼り用のカットバッファ（文字列の作業用一時格納エリア）が用意されており、テキストのブロック移動やブロックコピーなどを行うことができます。
● テキストの編集中に、他のファイルを読み込み、これを挿入したり追加したりすることができます。
● 迅速な処理を実現するため、確認のためのメッセージは極力少なく抑えてあります。
● 日本語入力を行うことも可能です。日本語の入力方法についての詳細は、『日本語入力・辞書ユーティリティユーザーズマニュアル』を参照してください。

1.3　EDの起動

EDを起動するには次のように入力します。

ED 〔<d:>〕〔<パス名>〕<ファイル名>〔.<拡張子>〕 ⏎

ここで、<d:>はドライブ名を、<パス名>はパス名を、<ファイル名>は主ファイル名を、<拡張子>は拡張子を、それぞれ示しています。

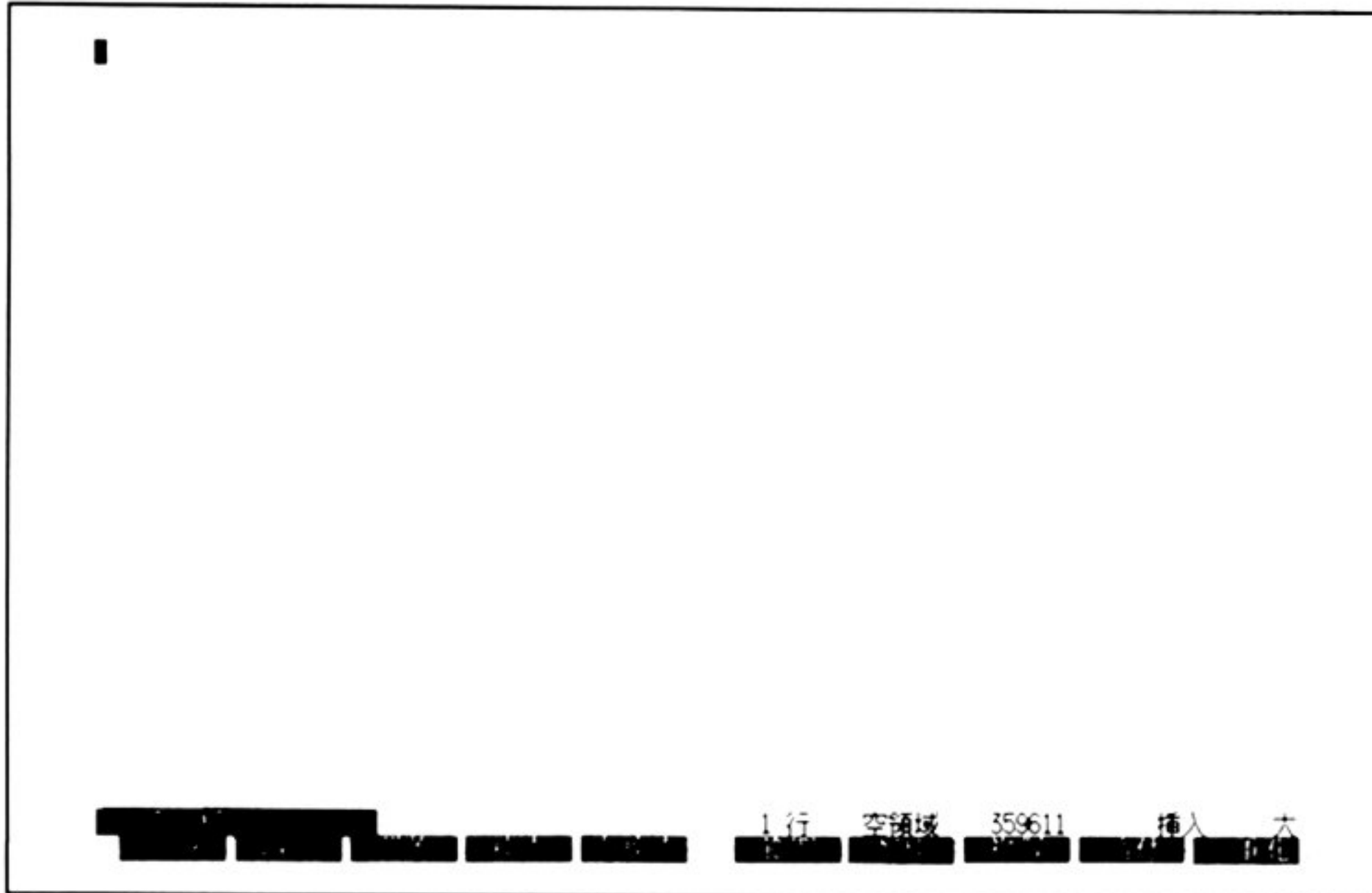
このときドライブ名を指定しないとカレントドライブが対象となります。また、何もファイル名を指定しないと "ED.$$$" というファイル名をもつファイルが自動的に作成されます。

"ED.$$$" ファイルが作成されたときは、EDの終了後、リネーム（REN）をするか、または COPY コマンドでその内容を他のファイルにコピーするようにしてください。

なお、EDの起動時には、さらにいろいろなスイッチを指定することもできます。これについての詳

細は、「第3章　EDのスイッチ付き起動」をご覧ください。
　次のように入力して、

ED　EDIT.TXT ⏎

EDを起動すると画面は下のようになります。

　ここで、画面の下部にあるメッセージは、左から順に、
- ●編集するファイル名
- ●カーソルのある行（現在行）の行番号（ファイルの先頭から何行目か）
- ●メモリの空き領域（バイト数）
- ●入力モード（入力された文字が挿入される〝挿入〟モードか、上書きされる〝上書〟モードか）
- ●大文字／小文字変換（CTRL＋］で、〝大〟文字に変換するか、〝小〟文字に変換するか）

を、それぞれ表しています。
　ここでは、ファイルにEDIT.TXTを指定して起動したため、編集するファイル名はEDIT.TXTとなっています。
　何かキーを押し続けてみてください。画面の右端を超えても文字を入力することができます。
　たとえば、〝A〟のキーを続けて入力すると、〝A〟が画面の右端に達したときに、さらに入力できるように画面全体が左方向に動きます。

AAAAAAAAAAAAAAAAAAAAAAAAAA～AAAAA

　このような機能を、横スクロールといいます。画面に表示できる1行の文字数は96文字（または64文字）ですが、横スクロールを利用することにより、行の折り返しなしで、最大文字数（128／256／512／1024のいずれか）まで入力することができます。
　適当なところでリターンキーを押すと、2行目の先頭にカーソルが移動します。

AAAAAAAAAAAAAAAAAAAAAAAAAA～AAAA ⏎
■

このとき、横スクロールしていた画面はもとの位置に戻ります。

同様にして、いろいろなキーを押して対応する文字が画面に表示されるのを確認してください。具体的なEDの使い方については第2章を参照してください。

1．4　EDの終了

(1)　ファイルの編集内容をセーブして終了

編集内容をディスクにセーブ（格納）して編集を終了したいときは、次のようにしてください。

[ESC]・[E]（[ESC]を押し、続けて[E]を押す）

編集中のファイルは、指定したファイル名でディスクにセーブされ、Human68kのコマンドモードに戻ります。

(2)　ファイルを編集前状態で終了

編集前の状態、すなわちEDの起動時に指定したファイルの内容を変更することなく終了したいときは、次のように入力してください。

[ESC]・[Q]（[ESC]を押し、続けて[Q]を押す）

すると、

本当にエディタを終了してよろしいですか（Y／N）

と表示されますので、ここで[Y]を押せば編集を終了します。[N]を押せば編集画面に戻ります。

第2章　EDのコマンドの解説

本章では、テキストの編集を行うためのコマンドやキー操作について、説明します。

コマンドには、CTRL (コントロールキー) 系とESC (エスケープキー) 系、それに単独キー系の3種類があります。次にその操作方法と本書での表記法について説明します。

◆ CTRL系のコマンド

CTRL系のコマンドは、CTRL を押しながら、アルファベット1文字のキーを押すことによって入力するものです。この操作方法を、

CTRL + X (×はアルファベット1文字のキー)

のようにプラス記号でつないで表示します。

◆ ESC系のコマンド

ESC系のコマンドは、まず ESC を押し、画面左下にアスタリスク(*)が表示されてから、続けてアルファベット1文字などのキーを押すことによって入力するものです。この操作方法を、

ESC・X (×は1文字のキー)

のように〝・〟記号で区切って表示します。

◆単独キー系のコマンド

単独キー系のコマンドは、ファンクションキーや INS 、DEL といった、単独の特殊キーで入力するものです。これには、SHIFT を押しながら用いる場合も含みますが、このときは、

SHIFT + X (×は単独の特殊キー)

のように表示します。

以降では、EDのコマンドの使用法を目的別に述べますが、それぞれの項で、この3種類のコマンド系に分けて解説します。

2.1　カーソルの移動

EDは、編集作業を画面上で行いますので、作業の中心は常にカーソル位置になります。文字の検索・削除や、行の削除・挿入などのような編集作業は、カーソルが表示されている位置を基準点として行われます。したがって、まず何かの作業を行う前には、カーソルを目的のところに移動しなくてはなりません。

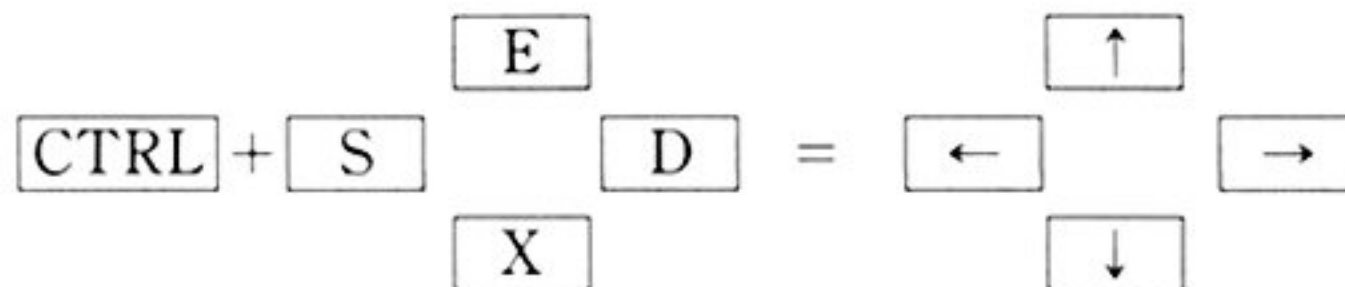
カーソルの移動を行うコマンドは、次のとおりです。

◆ CTRL 系のコマンド

● 1 文字単位の移動

CTRL + E	カーソルを上に移動します。
CTRL + X	カーソルを下に移動します。
CTRL + S	カーソルを左に移動します。
CTRL + D	カーソルを右に移動します。

この 4 つのコマンドは次のようにカーソル移動キーに対応しています。

```
              E                ↑
CTRL + S           D   =   ←        →
              X                ↓
```

●行単位の移動

CTRL + B	カーソルを行の先頭に移動します。 カーソルが行の先頭にあるときは行の終りに移動します。 それ以外は、行の先頭に移動します。
CTRL + Q	カーソルを行の左端に移動します。
CTRL + P	カーソルを行の右端に移動します。
CTRL + I	水平タブで指定した分カーソルを右に移動します。画面の右端にきたときは次の行に改行されます。
CTRL + M	改行します。挿入モードのときは、行を分割します。 ↵ と同じ機能です。

●ワード単位の移動

CTRL + A	カーソルを 1 ワード左に移動します。
CTRL + F	カーソルを 1 ワード右に移動します。 この 2 つのコマンドにより、カーソルをワード単位で移動することができます。ここで、ワードとは、空白、カンマ、水平タブなどで区切られた文字列の単位のことを示します。

●ページ単位の移動

CTRL + R	カーソルを 1 ページ前に移動します。 ROLL DOWN と同じ機能です。
CTRL + C	カーソルを 1 ページ後に移動します。 ROLL UP と同じ機能です。

◆ ESC 系のコマンド

ESC・B	カーソルを現在編集中のファイルの先頭に移動します。

ESC・Z	カーソルを現在編集中のファイルの終りに移動します。
ESC・n・↵	カーソルをn(数値)で指定した行に移動します。カラム位置(横方向の桁位置)は変わりません。

◆単独キー系のコマンド

HOME	カーソルをホーム位置(画面の左上)に移動します。
ROLL UP	カーソルを1ページ後に移動します。
ROLL DOWN	カーソルを1ページ前に移動します。
F 1	カーソルを現在編集中のファイルの先頭に移動します。
F 2	カーソルを現在編集中のファイルの終りに移動します。
↵	改行します。挿入モードのときは、行を分割します。
↑	カーソルを上に移動します。
↓	カーソルを下に移動します。
←	カーソルを左に移動します。
→	カーソルを右に移動します。
TAB	水平タブで指定した分カーソルを右に移動します。画面の右端にきたときは次の行に改行されます。

2.2　挿入

挿入を行うコマンドは、次のとおりです。

◆ CTRL 系のコマンド

CTRL+N	カーソルのある行(現在行)の前の行に空白行を1行挿入します。 現在行のカラム位置にかかわらず実行されます。

◆単独キー系コマンド

F10	現在行の次の行に現在行を挿入コピーします。
INS	キー上の赤のランプがついているときは挿入モードです。 文字は、カーソルのある位置の直前のカラムに挿入されます。EDの画面には〝挿入〞と表示されます。 ランプが消えているときは上書モードで、文字は、カーソルのある位置から上書きされます。EDの画面には〝上書〞と表示されます。 挿入と上書は〝トグル〞になっており、押すごとに状態が反転するようになっています。

2．3　削除

削除を行うコマンドは、次のとおりです。

◆ CTRL 系のコマンド

CTRL + G	カーソル位置の文字を1文字削除します。 DEL と同じ機能です。
CTRL + H	カーソル位置の前の文字を削除します（バックスペース）。 BS と同じ機能です。
CTRL + T	カーソル位置のワードを1ワード削除します。
CTRL + Y	現在行を1行削除します。 カーソルのカラム位置にかかわらず実行されます。 このキーで削除された文字列は、CTRL + L で復帰（挿入）することができます。
CTRL + U	行の先頭からカーソル位置の前までを削除します。
CTRL + K	カーソル位置の文字から行の終りまでを削除します。

2．4　検索

検索を行うコマンドは、次のとおりです。

◆ ESC 系のコマンド

ESC・N	文字列をカーソル位置から文末に向かって検索します。 コマンドの実行方法は、F 4 と同じです。
ESC・S	文字列をカーソル位置から文頭に向かって検索します。 コマンドの実行方法は、SHIFT + F 4 と同じです。

◆単独キー系のコマンド

F 4

文字列をカーソル位置から文末に向かって検索します。
指定した文字列をカーソル位置からさがし、最初に見つかったところにカーソルが移動します。
検索は1回のみ行われます。同じ文字列について、続いて検索を行いたいときには、F 5 を押してください。
F 4 を押すと画面左下に次のメッセージが表示されます。

前方検索文字列 ：■

■が点滅します。検索する文字列を入力し、⏎を押してください。
検索文字列と同じ文字列の先頭にカーソルが移動します。

該当する文字列が見つからないときは、〝文字列が見つかりません〟というメッセージが表示され、検索前の状態に戻ります。

SHIFT ＋ F 4

F 4 の逆方向の検索を行います。カーソル位置から文頭に向かって検索します。
コマンドの実行方法は F 4 と同じです。
検索は1回のみ行われます。同じ文字列について、続いて検索を行いたいときには、SHIFT ＋ F 5 を押してください。

F 5

指定した文字列を連続的に文末に向かって再検索します。
F 5 を用いる前に、F 4 または SHIFT ＋ F 4 で、検索文字列を指定しておく必要があります。
再検索は該当する文字列が見つからなくなるまで行われます。
F 5 を押すと指定した文字列の先頭にカーソルが移動します。
続いて F 5 を押すと次の文字列が再検索されます。

SHIFT ＋ F 5

F 5 の逆方向の再検索を行います。カーソル位置から文頭に向かって再検索します。
SHIFT ＋ F 5 を用いる前に、F 4 または SHIFT ＋ F 4 で、検索文字列を指定しておく必要があります。
コマンドの実行方法は F 5 と同じです。

2.5　置換

置換を行うコマンドは、次のとおりです。

◆ESC系コマンド

ESC ・ J

カーソル位置から文末に向かって、指定した文字列を新しい文字列に連続置換します。
コマンドを実行すると次のメッセージが表示されます。

前方置換旧文字列：■

ここで、置きかえたい（置換前）文字列を入力し、↵ を押してください。
続いて、次のメッセージが表示されます。

前方置換新文字列：■

新しく置きかえる（置換後）文字列を入力し、↵ を押してください。すると、

確認しますか？（Y／N）

とたずねてきます。ここで、ひとつひとつ確認しながら置換する場合はYを、確認せずにいっきに置換する場合はNを押します。

Yを押して該当する置換前文字列が見つかると、

置換しますか？（Y／N）

とたずねてきます。ここで、置換を行う場合はYを、置換しない場合はNを押します。
続いて、次の置換前文字列がさがされます。置換は、置換前文字列がなくなるまで続けられます。
置換を中止し、編集画面に戻りたい場合は、ESCを押してください。

ESC・L

カーソル位置から文頭に向かって、指定した文字列を新しい文字列に連続置換します。
コマンドを実行すると次のメッセージが表示されます。

後方置換旧文字列：■

ここで、置きかえたい（置換前）文字列を入力し、⏎を押してください。
続いて、次のメッセージが表示されます。

後方置換新文字列：■

新しく置きかえる（置換後）文字列を入力し、⏎を押してください。

確認しますか？（Y／N）

とたずねてきます。ここで、ひとつひとつ確認しながら置換する場合はYを、確認せずにいっきに置換する場合はNを押します。
Yを押して該当する置換前文字列が見つかると、

置換しますか？（Y／N）

とたずねてきます。ここで、置換を行う場合はYを、置換しない場合はNを押します。
続いて、次の置換前文字列がさがされます。置換は、置換前文字列がなくなるまで続けられます。
置換を中止し、編集画面に戻りたい場合は、ESCを押してください。

ESC・R

カーソル位置から文末に向かって、指定した文字列を新しい文字列に連続置換します。置換の確認動作はしません。
コマンドを実行すると次のメッセージが表示されます。

前方置換旧文字列：■

ここで、置きかえたい（置換前）文字列を入力し、⏎を押してください。
続いて、次のメッセージが表示されます。

前方置換新文字列：■

新しく置きかえる（置換後）文字列を入力し、⏎を押してください。
すると、

検索中　ESCで中止

と表示され、該当する置換前文字列はすべて置換後文字列に置換されます。
置換を中止し、編集画面に戻りたい場合は、ESCを押してください。

ESC・U

カーソル位置から文頭に向かって、指定した文字列を新しい文字列に連続置換します。置換の確認動作はしません。
コマンドを実行すると次のメッセージが表示されます。

後方置換旧文字列：■

ここで、置きかえたい（置換前）文字列を入力し、⏎を押してください。
続いて、次のメッセージが表示されます。

後方置換新文字列：■

新しく置きかえる（置換後）文字列を入力し、⏎を押してください。
すると、

検索中　ESCで中止

と表示され、該当する置換前文字列はすべて置換後文字列に置換されます。
置換を中止し、編集画面に戻りたい場合は、ESCを押してください。

◆単独キー系コマンド

F3

カーソル位置から文末に向かって、指定した文字列を新しい文字列に連続置換します。置換の確認動作はしません。
コマンドの実行方法はESC・Rと同じです。

SHIFT＋F3

カーソル位置から文頭に向かって、指定した文字列を新しい文字列に連続置換します。置換の確認動作はしません。
コマンドの実行方法はESC・Uと同じです。

2.6 ファイルの読み込み・書き出し

ファイルの読み込み・書き出しを行うコマンドは、次のとおりです。

◆ESC系のコマンド

ESC・Y

別のファイルを編集中のファイルに読み込みます。
すでに作成してある別のファイルを読み込むときに使用します。現在行(カーソルのある行)の直前に、指定したファイル全体が挿入されます。コマンドを実行すると画面左下に次のメッセージが表示されます。

ファイル読込み：■

ここで読み込むファイルのファイル名を入力します。続いて↵を押してください。指定ファイルが読み込まれます。
中止して編集画面に戻る場合には、ESCを押してください。

ESC・W

編集中のファイルから一部分をとり出し、新しいファイルとして作成(書き出し)します。
これには、2通りの方法があります。
(1) 現在行からファイルの最後までを書き出す場合
ESC・Wをそのまま実行すると、次のようなメッセージが表示されます。

ファイル書出し：■

ここで、作成するファイルのファイル名を入力し、↵を押してください。
中止して編集画面に戻る場合には、ESCを押してください。
(2) 特定範囲内の内容を書き出す場合
まず、指定したい範囲の最初の行にカーソルを移動してから、F 6を押し、範囲指定を開始します。
続いて、指定したい範囲の最終行の次の行にカーソルを移動します。次に、ESC・Wを実行すると、次のメッセージが表示されます。

ファイル書出し：■

ここで、作成するファイルのファイル名を入力し、↵を押してください。
中止して編集画面に戻る場合には、ESCを押してください。

2.7　カットバッファ

カットバッファはまとまった量の文字列を削除・複写・貼付するときに使用します。
〝カット〟バッファは、あるまとまりのある文字列を切り取って（カット）一時的に格納しておき、次の処理（移動・複写）に備えるようになっているものです。

◆ESC系コマンド

ESC・n・P ↵　カーソルのある行からn行分カットバッファに格納します。n行分のテキストは削除されます。nは任意の数値で、これを省略すると、1となります。

ESC・n・G ↵　カットバッファの内容を現在行の直前にn回挿入します。nは任意の数値で、これを省略すると、1となります。

◆単独キー系コマンド

F6　カットバッファにテキストを格納する前に、格納する範囲を行単位で指定するために使います。
指定したい範囲の最初の行にカーソルを移動してから F6 を押すと、画面左下に次のように表示されます。

範囲指定

続いて、指定したい範囲の最終行の次の行にカーソルを移動し、次に説明する F7 あるいは F8 を押します。

F7　F6 で指定した行から現在行の前の行にいたるまでの範囲内のテキストを、カットバッファに格納します。指定範囲内のテキストは削除されます。

F8　F6 で指定した行から現在行の前の行にいたるまでの範囲内のテキストを、カットバッファに格納します。指定範囲内のテキストはそのまま削除されずに残ります。

F9　カットバッファの内容を現在行の直前に挿入します。

カットバッファを使用して実際にテキストの編集をするときは、次のようにします。

●テキストの移動

1．移動元となるテキストの最初の行にカーソルを移動し、F6 を押して範囲指定を開始します。
2．次に、移動元のテキストの最終行の次の行にカーソルを移動し、ここで F7 を押して移動元テキスト全体をカットバッファに格納します。このとき、指定したテキストは、編集画面上から削除されます。
3．カーソルを移動して、カットバッファ中のテキストを挿入したい場所までもっていき、ESC・n・G ↵ 実行するか、または F9 を押します。（ESC・n・G ↵ では、nに数値を指定すれば、n回分挿入できます）。

4．テキストは、現在行の直前に挿入され、結果的にテキストが移動されたことになります。

なお、1．～2．の作業の代わりに、[ESC]・[n]・[P][↵]を使って指定行分のテキストをカットバッファに格納することもできます。

●テキストの複写

1．複写元となるテキストの最初の行にカーソルを移動し、[F 6]を押して範囲指定を開始します。

2．次に、複写元のテキストの最終行の次の行にカーソルを移動し、ここで[F 8]を押して複写元テキスト全体をカットバッファに格納します。このとき、指定したテキストは、編集画面上からは削除されずに残ります。

3．カーソルを移動して、カットバッファ中のテキストを挿入したい場所までもっていき、[ESC]・[n]・[G][↵]を実行するか、または[F 9]を押します。（[ESC]・[n]・[G][↵]では、nに数値を指定すれば、n回分挿入できます。）

4．テキストは、現在行の直前に挿入され、結果的にテキストが複写されたことになります。

なお、1．～2．の作業の代わりに、[ESC]・[n]・[P]・[↵]を使って指定行分のテキストをカットバッファに格納することもできます。ただし、画面上からは削除されます。

2．8　その他の編集コマンド

◆CTRL系コマンド

[CTRL]＋[J]

EDのヘルプファイルを表示します。
コマンドの使い方を参照するときに使用します。
[HELP]と同じ機能です。

[CTRL]＋[ ]]

現在のカーソルの位置の文字を、大文字または小文字に変換します。
大文字に変換するか小文字に変換するかは、[ESC]・[ ]]で切りかえられます。

[CTRL]＋[V]

コントロールコードを入力します。
[CTRL]＋[V]に続けて、入力したいコントロールコードの大文字を押します。

◆ESC系コマンド

[ESC]・[O]

現在編集中のファイルを編集前の状態に戻します。それまでに行った編集作業は無効となり、最初の状態となります。
コマンドを実行すると、

再編集（Y／N）

とたずねてきます。[Y]を押せば編集前の状態に戻り、[N]を押せば以前の編集画面に戻ります。

ESC・T

現在編集中のファイルのファイル名を変更します。
コマンドを実行すると画面左下に次のメッセージが表示されます。

ファイル名変更：××××.×××■

××××.×××には、現在編集中のファイル名が表示されます。
ここで、BSを使ってカーソルを前に戻し、新しいファイル名を入力後、↵を押してください。

ESC・C

EDから抜け出すことなくHuman68kのコマンドを実行するときに使用します。
実行すると画面左下に次のメッセージが表示されます。

Human68k：■

■にHuman68kのコマンドを入力します。コマンドが正しいときは指定したコマンドが実行された後、任意のキーの入力待ちとなります。何かキーを押すと、EDの編集に戻ります。また、ここで単に↵だけを押すと、Human68kのコマンド待ちになります。再びEDに戻るときはEXIT↵と入力します。

ESC・M

改行記号（↓）を表示するときに使用します。
EDでは通常、改行文字は表示されません。このコマンドを実行すると"↓"記号により改行の位置が表示されます。
もう一度ESC・Mを実行すると、改行記号の表示は解除されます。

ESC・]

大文字・小文字変換モードの切りかえをします。
CTRL＋]を使って、現在カーソルのあるところの文字を大文字または小文字に変換するときに、どちらに変換するのかを切りかえるのに使います。
大文字変換モードになっているときには、画面右下に"大"と表示され、小文字変換モードになっているときには"小"と表示されます。

ESC・I

タブ記号（→・・・）を表示するときに使用します。
EDでは通常、水平タブは表示されません。このコマンドを実行すると"→・・・"記号によりタブの位置が表示されます。
もう一度ESC・Iを実行すると、タブ記号の表示は解除されます。

ESC・@

マクロを定義します。
ESC・@を押して、次にESC・@を押すまでに入力した内容がマクロとして定義されます。定義したマクロはUNDOを押すと実行されます。

◆単独キー系コマンド

HELP	EDのヘルプファイルを表示します。
CLR	ファイルの終端記号（EOF）を表示するときに使用します。 EDでは通常、ファイル終端記号は表示されません。このコマンドを実行すると〝EOF〞記号によりファイルの終端位置が表示されます。 もう一度 CLR を押すと、ファイル終端記号の表示は解除されます。
UNDO	定義されたマクロを実行します。（ESC・@ 参照）。

2.9　ファイルのセーブ・編集の終了

◆ESC系コマンド

ESC・H	現在編集中のファイルだけをディスクにセーブします。 現在の編集状態を一度記録するときに使用します。 コマンドを実行しても、画面上では何も起こりませんが、現在編集中のファイルだけが、現在のファイル名でセーブされます。 なお、ファイル名を変更してセーブしたい場合は、あらかじめ ESC・T でファイル名の変更をしておいてから、このコマンドを実行してください。
ESC・K	現在編集中のファイルだけを編集前の状態で終了します。 複数ファイルの編集（2.10節参照）を行っている場合、現在編集中のファイルだけを終了し、残りのファイルの編集は継続します。
ESC・X	現在編集中のファイルだけをディスクにセーブした後、EDを終了します。 複数ファイルの編集（2.10節参照）を行っている場合、現在編集中のファイルだけをセーブし、これ以外のファイルはセーブしません。 なお、ファイル名を変更してセーブしたい場合は、あらかじめ ESC・T でファイル名の変更をしておいてから、このコマンドを実行してください。
ESC・E	編集中のすべてのファイルをディスクにセーブした後、EDを終了します。 複数ファイルの編集（2.10節参照）を行っている場合、編集中のすべてのファイルをセーブします。 なお、ファイル名を変更してセーブしたい場合は、各ファイルにつき、あらかじめ ESC・T でファイル名の変更をしておいてから、このコマンドを実行してください。
ESC・Q	編集中のすべてのファイルを編集前の状態で終了します。 それまで編集した結果をすべて無効にするときに使用します。

コマンドを実行すると画面左下に次のメッセージが表示されます。

本当にエディタを終了してもよろしいですか（Y／N）

ここで[Y]を押すと、ファイルの編集前の状態でEDを終了します。[N]を押すと、もとの編集画面に戻ります。

2.10　複数ファイルの編集

EDでは、一度に10個までのファイルを編集することができます。

複数ファイルの編集を行う場合には、コマンドモードからEDを起動する際に、次のように指定します。各ファイルのファイル名は、スペースで区切ってください。

ED　〔<ファイル1>〕〔<ファイル2>〕………〔<ファイル10>〕

この形式でEDを起動した後、次に示すようなコマンドを用いることにより、編集の対象となるファイルを切りかえながら、それぞれのファイルについて編集を行うことができます。

◆ESC系コマンド

[ESC]・[A]

編集の対象となるファイルの切りかえを行います。

ファイルの切りかえは、ED起動時に指定した複数ファイルの順、すなわち、

ファイル1→ファイル2→
・・・→ファイル9→ファイル１０→ファイル1

の順に行われます。

[ESC]・[D]

編集の対象となるファイルの切りかえを行います。

ファイルの切りかえは、ED起動時に指定した複数ファイルの逆順、すなわち、

ファイル1→ファイル１０→ファイル9→
・・・→ファイル2→ファイル1

の順に行われます。

[ESC]・[F]

新しいファイルを作成します。

コマンドを実行すると、

編集ファイル：　■

と表示されます。作成したいファイルのファイル名を入力し、[↵]を押してください。新しいファイルが編集の対象となります。

ESC・V　　タグジャンプを行います。
編集中、他のファイルを参照したい場合に使うもので、次のようにして実行します。
まず、つぎの例のように、編集画面中の任意の1カラム目から、参照したいファイルのファイル名を入力し、続けて TAB を押してから数値を入力します。

CONFIG. SYS TAB 3

数値は、目的のファイルが開かれたときの、カーソルの位置（行数）を示します。
続いて、ESC・V を実行してください。この例では、"CONFIG. SYS" ファイルが開かれた後、3行目にカーソルが置かれます。
なお、指定のファイルが見つからない場合は、新しいファイルとして開きます。
また、FINDコマンドでFIND text ＊.S ／F ＞ TEMPと実行し、その出力をリダイレクト機能を用いて、TEMPというファイルに作成し、そのファイルでタグジャンプを使ってすばやく参照することができます。

◆単独キー系コマンド

SHIFT＋F6　　編集の対象となるファイルの切りかえを行います。
ファイルの切りかえは、ED起動時に指定した複数ファイルの順に行われます。
ESC・A と同じ機能です。

SHIFT＋F7　　編集の対象となるファイルの切りかえを行います。
ファイルの切りかえは、ED起動時に指定した複数ファイルの逆順に行われます。
ESC・D と同じ機能です。

なお、複数ファイルの編集を終了する場合、すべてのファイルについて終了したければ、次のいずれかを使用しなくてはなりません。

ESC・E　　編集中のすべてのファイルをディスクにセーブした後、EDを終了します。
ESC・Q　　編集中のすべてのファイルを編集前の状態で終了します。

次のコマンドを使用した場合は、現在編集中のファイルについてだけしか終了しませんので注意が必要です。

ESC・X　　現在編集中のファイルだけをディスクにセーブした後、EDを終了します。
ESC・K　　現在編集中のファイルだけを編集前の状態で終了します。

第3章　EDのスイッチ付き起動

EDはスイッチを付けて起動することにより、その実行環境の設定を行うことができます。
EDをスイッチ付きで起動するには、次の書式を使います。

ED 〔−A〕〔−E〕〔−H〕〔−L〕〔−M〕〔−S〕〔−T〕
〔<ファイル1>〕…〔<ファイル10>〕

〈ファイル1〉〜〈ファイル10〉については、「2.10　複数ファイルの編集」に説明してありますので、参照してください。
2つ以上のスイッチを付けるときには、スイッチとスイッチのあいだにスペースを入れてください。

各スイッチの意味は、次に示すとおりです。

−A　ヘルプファイルのパス指定。
　ヘルプファイルの存在しているディレクトリのフルパス名で指定します。
　デフォルトでは、ヘルプファイルのパス名は、ED.Xの存在しているディレクトリのパス名と同じとなります。

−E　ファイル終端記号（EOF）の表示指示。
　デフォルトでは、EOF記号は表示されません。

−H　水平タブの最大表示幅指定。
　水平タブ（＄09）の最大表示幅をカラム数で指定します。
　このスイッチの直後に、
　　2、4、6、8
　のうちのどれかを指定します。
　デフォルトでは、8カラムに指定されています。

−L　改行記号（↓）の表示指定。
　デフォルトでは、改行記号は表示されません。

−M　行長の指定。
　1行の長さをバイト数で指定します。
　このスイッチの直後に、
　　128、256、512、1024
　のうちのどれかを指定します。
　デフォルトでは、512バイトに指定されています。

－S　画面モードの指定。

このスイッチの直後に、モードを設定する数値を指定します。

1…96（カラム）×30（行）

2…64（カラム）×30（行）

デフォルトは、1です。

－T　タブ記号（→・・・）の表示指定。

デフォルトでは、タブ記号は表示されません。

第４章　EDの操作例

ここでは、EDの簡単な操作例として、小さなファイルの作成を行う方法を示します。
例として作成するのは、簡単なバッチファイルですが、同様な方法で、プログラムやメモ書きなども作成することができます。
まず、Human68kのコマンドモード（入力待ち状態）から次のように入力します。

ED　LPR.BAT ↵

ここで、"LPR.BAT" が、作成するファイル名になります。すでにLPR.BATが作られているときはその内容が、そうでないときは何も入力されていない、新たな画面が表示されます。
ここでは、LPR.BATファイルを新たに作成するものとして説明します。
では、次のような内容のLPR.BATファイルを作ってみます。

```
ECHO OFF
IF "%1" == "" GOTO NONFILE
IF NOT EXIST %1 GOTO NONFILE
PR /F /H"%1" /W80 /L60 %1 PRN
GOTO END
:NONFILE
ECHO !!!!!!  警告  !!!!!!
ECHO ファイルがありません
ECHO !!!!!!!!!!!!!!!!!!!!
:END
```

実際に入力してみましょう。文字を大文字で入力するときは、CAPSの赤ランプがついている状態にします。
まず、

ECHO　OFF

を入力してみます。
"ECHO" に続いてスペース(空白)を入力します。スペースはスペースキーを押すと入力できます。
もし、入力まちがいをしたら、BSまたはCTRL＋Hを押すと、カーソルの直前の文字を消すことができます。またDELまたはCTRL＋Gを押せば、カーソルのところにある文字を消せます。
ここで改行をします。改行は、リターンキーを押すかCTRL＋Mを押します。改行されたかどうか確認するために、ESC・M（ESCを押してから、Mを押す）を実行してみてください。
改行記号（↓）が行末に表示されていれば正しく改行されています。

次に、

```
IF "%1" == "" GOTO NONFILE
```

を入力します。

これは、前の行と同じ要領で行うことができます。

同様にして、

```
:NONFILE
```

まで入力してください。続いて、

```
ECHO !!!!!! 警告 !!!!!!
ECHO ファイルがありません
ECHO !!!!!!!!!!!!!!!!!!!!
```

を入力します。

ここで、〝ECHO 〞を3行連続して入力することになりますが、これを能率よく行うには、最初に〝ECHO 〞を入力してから、カーソルをその行に置いたまま [F10] を2回押します。〝ECHO 〞が次の2行にコピーされ、3行連続して同じ文字列が入力されます。

次にスペース以降を入力します。そして [←] および [→] (または [CTRL]+[S] および [CTRL]+[D])でカーソルを左右に動かし、〝!!!!!! 警告 !!!!!!〞を入力します。

次に、[↓] または [CTRL] + [X] を押してカーソルを下に動かし、〝ファイルがありません〞、〝!!!!!!!!!!!!!!!!!!!!〞を順に入力します。最後に〝:END〞を入力します。なお、日本語の入力方法は、『日本語入力・辞書ユーティリティユーザーズマニュアル』を参照してください。

入力まちがいをして、1行全体を消したいときには、[CTRL] + [Y] を押してください。

入力を終えたら誤りがないかどうか確認します。正確に入力できたならば、LPR.BAT ファイルの内容をディスクにセーブします。これには [ESC]・[H] を実行します。すると、編集画面には何の変化も起こりませんが、セーブが行われます。

ファイルをセーブした後、ED を終了したければ、[ESC]・[E] を実行します。セーブの終了後、Human68k のコマンドモード (入力待ち状態) に戻ります。

なお、最初のEDの起動時に、パラメータとして〝LPR.BAT〞を指定せず、

```
ED ↵
```

としてファイルの内容を入力した後、セーブする前にファイル名を LPR.BAT に変更することもできます。

これを行うためには、[ESC]・[T] を実行します。現在編集中のファイル名として〝ED.$$$〞が表示されますから、[BS] を使ってこれを消し、新しいファイル名 (LPR.BAT) を入力してから、リターンキーを押します。これで、現在編集中のファイル名が〝LPR.BAT〞に変わりますから、以後、セーブを行えば、この新しいファイル名でファイルが作成されます。

これまでの説明に出てきたコマンドは ED が持っている中のごく一部です。このほかにも便利で強力なコマンドが数多くあります。これらのコマンドは、「第5章 EDのコマンド一覧」にすべて示されていますので、参照しながら実行してみてください。

第5章　EDのコマンド一覧

これまでの章でEDの操作をひととおり解説しました。この章では、EDで使うことのできる全コマンドをまとめてありますので、利用してください。

(1) CTRL系コマンド

キー	機能
CTRL+A	カーソルを1ワード左に移動します
CTRL+B	カーソルを行の右端・左端に移動します
CTRL+C	ページアップします(1ページ後に移動)
CTRL+D	カーソルを1文字右に移動します(→と同じ)
CTRL+E	カーソルを1行上に移動します(↑と同じ)
CTRL+F	カーソルを1ワード右に移動します
CTRL+G	1文字削除します(DELと同じ)
CTRL+H	バックスペース(BSと同じ)
CTRL+I	タブ(TABと同じ)
CTRL+J	ヘルプファイルを示します(HELPと同じ)
CTRL+K	カーソル位置から行末までを削除します
CTRL+L	CTRL+Yで削除した文字列を現在行に復帰(挿入)します
CTRL+M	改行。挿入モードでは、行を分割します(リターンキー↵と同じ)
CTRL+N	1行挿入します
CTRL+O	挿入・上書のモードを切りかえます
CTRL+P	カーソルを行の右端に移動します
CTRL+Q	カーソルを行の左端に移動します
CTRL+R	ページダウンします(1ページ前に移動)
CTRL+S	カーソルを1文字左に移動します(←と同じ)
CTRL+T	1ワード削除します
CTRL+U	行の先頭からカーソル位置の直前までを削除します
CTRL+V	コントロールコードを入力する前に入力します
CTRL+W	1行分ロールダウンします
CTRL+X	カーソルを1行下に移動します(↓と同じ)
CTRL+Y	1行削除します
CTRL+Z	1行分ロールアップします
CTRL+]	カーソル位置の文字を大文字または小文字に変換します

(2) ESC系コマンド

キー	機能
ESC・A	複数ファイル編集時のファイルの切りかえ(正順)
ESC・B	カーソルをファイルの先頭に移動します(F1と同じ)

キー	機能
ESC・C	Human68k のコマンドを実行します
ESC・D	複数ファイル編集時のファイルの切りかえ（逆順）
ESC・E	編集中のファイルすべてをセーブして ED を終了します
ESC・F	新しいファイルの編集を行います
ESC・n・G ↵	カットバッファの内容を現在行の直前に n 回挿入します
ESC・H	現在編集中のファイルをセーブします。編集は継続します
ESC・I	タブ記号（→・・・）の表示／非表示を切りかえます
ESC・J	文末に向かって文字列を連続置換します（置換確認あり）
ESC・K	現在編集中のファイルだけを編集前の状態で終了します
ESC・L	文頭に向かって文字列を連続置換します（置換確認あり）
ESC・M	改行記号（↓）の表示／非表示を切りかえます
ESC・N	文字列を検索します（文末方向）
ESC・O	現在編集中のファイルを編集前の最初の状態に戻します
ESC・n・P ↵	現在行から n 行分カットバッファに格納します
ESC・Q	編集中のファイルすべてを編集前の状態で終了します
ESC・R	文末に向かって文字列を連続置換します（置換確認なし）
ESC・S	文字列を検索します（文頭方向）
ESC・T	現在編集中のファイルのファイル名を変更します
ESC・U	文頭に向かって文字列を連続置換します（置換確認なし）
ESC・V	タグジャンプを行います
ESC・W	編集中のファイルから一部をとり出し、ファイルを作成します
ESC・X	現在編集中のファイルだけをセーブし、ED を終了します
ESC・Y	編集中のファイルに、別のファイルを読み込みます
ESC・Z	カーソルをファイルの終りに移動します（F 2 と同じ）
ESC・n ↵	カーソルを第 n 行に移動します
ESC・]	大文字変換・小文字変換モードを切りかえます
ESC・@	マクロ定義を行ないます

(3)単独キー系コマンド

キー	機能
F 1（先頭行）	カーソルをファイルの先頭に移動します
F 2（最終行）	カーソルをファイルの終りに移動します
F 3（置換）	文末に向かって文字列を連続置換します（置換確認なし）
F 4（検索）	文字列を検索します（文末方向）
F 5（次検索）	文字列を再検索します（文末方向）
F 6（範囲）	行範囲の最初行を指定します
F 7（削除）	F 6 の指定行から現在行の前の行までのテキストをカットバッファに格納します 指定範囲内のテキストは削除されます
F 8（複写）	F 6 の指定行から現在行の前の行までのテキストをカットバッファに格納します

キー	説明
	指定範囲内のテキストは削除されません
F 9 (貼付)	カットバッファの内容を現在行の直前に挿入します
F10 (二重化)	現在行を次の行に挿入コピーします
SHIFT + F 3	文頭に向かって文字列を連続置換します(置換確認なし)
SHIFT + F 4	文字列を検索します(文頭方向)
SHIFT + F 5	文字列を再検索します(文頭方向)
SHIFT + F 6	複数ファイル編集時のファイルの切りかえ（正順)
SHIFT + F 7	複数ファイル編集時のファイルの切りかえ（逆順)
→	カーソルを1文字右に移動します
←	カーソルを1文字左に移動します
↑	カーソルを1行上に移動します
↓	カーソルを1行下に移動します
HOME	カーソルをホーム位置（画面左上）に移動します
INS	挿入・上書モードを切りかえます
DEL	1文字削除します
ROLL UP	カーソルを1ページ後に移動します
ROLL DOWN	カーソルを1ページ前に移動します
TAB	水平タブ
BS	バックスペース
HELP	ヘルプファイルを表示します
↵	改行。挿入モードでは、行を分割します
CLR	ファイル終了記号（EOF）の表示/非表示を切りかえます
UNDO	定義されたマクロを実行します

付　録

付録1　ASCII制御コード

次に示すASCIIコードをコンソール(画面)に対して出力することにより、画面を制御することができます。制御コードは1文字です。

記号	コード(16進)	機　　能
BEL	07	Sound bell ベルを鳴らします。 ベルの音は、CONFIG.SYSファイルの中の"BELL=<ファイル名>"の記述に従います。"BELL=<ファイル名>"の指定がない場合は無効。
BS	08	Cursor backward カーソルを1文字左に移動します。 カーソルが行の1カラム目にあるときは1行上の最終カラムに移動します。 カーソルがホームポジション（1カラム目、1行目）にあるときは無効。
HT	09	Skip to next tab stop カーソルを次のタブの位置に移動します。 タブの位置は、キャラクタ座標の最大カラム数を越えない8の倍数で、次のように決められます。 08、16……キャラクタ座標の最大カラム数より小さい8の倍数の最大値(最大のタブ位置) カーソルが最大のタブ位置のカラムより右側にあるときは1行下の1カラム目に移動します。 カーソルが最終行にあるときは1行スクロールアップします。
LF	0A	Cursor down 同じカラム位置でカーソルを1行下に移動します。 カーソルが最終行にあるときは1行スクロールアップします。
VT	0B	Cursor up 同じカラム位置でカーソルを1行上に移動します。 カーソルが画面の1行目にあるときは無効。
FF	0C	Cursor forward カーソルを1文字右に移動します。 カーソルが行の最終カラムにあるときは1行下の1カラム目に移動します。 カーソルが画面最終行の最終カラムにあるときは1行スクロールアップします。
CR	0D	Cursor to left margin カーソルを行の1カラム目に移動します。

記号	コード (16進)	機　　能
SUB	1A	Clear screen 画面の表示をすべてクリアします。 その後、カーソルはホームポジションに移動します。
ESC	1B	Introduce on ESC sequence エスケープコードの始まりです。 ESC コードに続く文字により、エスケープシーケンスによる画面制御を行います(エスケープシーケンスの詳細については、「付録2　エスケープシーケンス」を参照)。
RS	1E	Cursor HOME カーソルをホームポジションに移動します。

付録2　エスケープシーケンス

エスケープシーケンスは、ESCコードとそれに続く文字列により画面の制御を行います。

エスケープシーケンスの長さは、シーケンスごとに異なります。ESCはエスケープコード（$1B）を表します。パラメータpn、pl、pc、psは10進数の文字を意味します。

シーケンス	機　　能
ESC [pl;pcH	Direct cursor addressing カーソルを指定の位置に移動します。 pl＝mのとき、mが最終行の値より大きい場合は最終行にカーソルを移動します。 pl＝0またはplが省略されたとき、1行目にカーソルを移動します。 pc＝nのとき、nが最終カラムの値より大きい場合は最終カラムに、pc＝0またはpcが省略された場合は1カラム目に、それぞれカーソルを移動します。
ESC [pl;pcf	ESC [pl;pcHと同様の処理を行います。
ESC＝lc	ESC [pl;pcHと同様の処理を行います。 パラメータl、cは、行・カラム位置を指定するもので、各パラメータは16進数で$20のオフセットを加算した文字によって表します。 l、cは省略不可。 lは行の位置を指定します。1行目にカーソルを移動するときは$20とします。 cはカラム位置を指定します。1カラム目にカーソルを移動するときは$20とします。
ESC [pnA	Cursor up 同じカラム位置でカーソルを上にpn行移動します。 カーソルが1行目にあるときは無効。 pnが省略されたとき、pn＝0のときはpn＝1として実行します。
ESC [pnB	Cursor down 同じカラム位置でカーソルを下にpn行移動します。 カーソルが最終行にあるときは無効。 pnが省略されたとき、pn＝0のときはpn＝1として実行します。
ESC [pnC	Cursor forward カーソルを右にpn文字移動します。 カーソルが行の最終カラムにあるときは無効。 pnが省略されたとき、pn＝0のときはpn＝1として実行します。

シーケンス	機　　能
ESC [pnD	Cursor backward カーソルを左に pn 文字移動します。 カーソルが行の1カラム目にあるときは無効。 pn が省略されたとき、pn＝0のときは pn＝1として実行します。
ESC [0J	Clear from cursor to end of screen カーソルの位置から最終行の最終カラムまでクリアします。 カーソルの位置は変更ありません。 パラメータ0は省略可能。
ESC [1J	Clear from beginning of screen to cursor 先頭行の1カラム目からカーソルの位置までをクリアします。 カーソルの位置は変更ありません。
ESC [2J	Clear screen 画面をすべてクリアします。 その後、カーソルはホームポジションに移動します。
ESC *	ESC [2J と同様。
ESC [0K	Clear from cursor to end of line カーソルの位置から行の最終カラムまでをクリアします。 カーソルの位置は変更ありません。 パラメータ0は省略可能。
ESC [1K	Clear from beginning of line to cursor 行の1カラム目からカーソルの位置までをクリアします。 カーソルの位置は変更ありません。
ESC [2K	Clear entire line containing cursor カーソルのある行の1カラム目から最終カラムまでをクリアします。 カーソルの位置は変更ありません。
ESC [pnM	Delete line カーソルのある行から下に pn 行削除します。 削除した行の後の行が繰り上がります。 カーソルの位置は繰り上がった文の先頭行の1カラム目となります。 pn が省略されたとき、pn＝0のときは pn＝1として実行します。

シーケンス	機　　能
ESC [pnL	Insert line カーソルのある行から下に pn 行空白行を挿入します。 カーソルのある行より下の行は pn 行分下にスクロールします。 カーソルの位置は挿入された先頭の空白行の 1 カラム目となります。 pn が省略されたとき、pn＝0 のときは pn＝1 として実行します。
ESC D	Index 同じカラム位置でカーソルを下に 1 行移動します。 カーソルが最終行にあるときは 1 行スクロールアップします。
ESC E	Next line カーソルを 1 行下の行の左端に移動します。 カーソルが最終行にあるときは 1 行スクロールアップします。
ESC M	Reverse index 同じカラム位置でカーソルを上に 1 行移動します。 カーソルが先頭行にあるときは 1 行スクロールダウンします。
ESC [s	Save cursor position カーソルの位置（行、カラム）とその表示文字の属性をセーブします。
ESC [u	Set cursor position カーソルを ESC [s でセーブした位置・属性に戻します。 ESC[s が実行されていないときは、カーソルはホームポジションに移動し属性は変化しません。
ESC [6n	Cursor position report 実行後、コンソール入力の度にカーソルの位置を ESC[pl;pcR の形で通知します。
ESC [>5l	Enable cursor display カーソルを画面上に表示します。 カーソルの位置は変更ありません。 Human68k システムの初期値。
ESC [>5h	Disable cursor display カーソルを画面上に表示しません。 カーソルの位置は変更ありません。 実行後、再び [>5l が出力されるまでカーソルは表示されません。

<table>
<tr><th>シーケンス</th><th>機　能</th></tr>
<tr><td>ESC [>1h</td><td>Enable bottom line
画面の最終行をプログラムで使用可能とします。
実行後、画面のすべての行をプログラム上で使用できます。</td></tr>
<tr><td>ESC [>1l</td><td>Disable bottom line
画面の最終行を Human68k システムで使用可能とします。
実行後、最終行はプログラム上で使用できません。</td></tr>
<tr><td>ESC [ps;…
;ps m</td><td>Character attribute
表示文字の属性を指定します。
属性は一度指定すると次に変更するまで有効となります。
パラメータ ps は一度に複数指定可能。
指定する ps を次に示します。
<table>
<tr><th>ps</th><th>機　能</th></tr>
<tr><td>0</td><td>初期設定時の属性</td></tr>
<tr><td>1</td><td>ハイライト</td></tr>
<tr><td>7</td><td>リバース</td></tr>
<tr><td>30</td><td>黒</td></tr>
<tr><td>31</td><td>水色</td></tr>
<tr><td>32</td><td>黄色</td></tr>
<tr><td>33</td><td>白</td></tr>
<tr><td>34</td><td>ハイライト 黒</td></tr>
<tr><td>35</td><td>ハイライト 水色</td></tr>
<tr><td>36</td><td>ハイライト 黄色</td></tr>
<tr><td>37</td><td>ハイライト 白</td></tr>
<tr><td>40</td><td>リバース 黒</td></tr>
<tr><td>41</td><td>リバース 水色</td></tr>
<tr><td>42</td><td>リバース 黄色</td></tr>
<tr><td>43</td><td>リバース 白</td></tr>
<tr><td>44</td><td>ハイライト・リバース 黒</td></tr>
<tr><td>45</td><td>ハイライト・リバース 水色</td></tr>
<tr><td>46</td><td>ハイライト・リバース 黄色</td></tr>
<tr><td>47</td><td>ハイライト・リバース 白</td></tr>
</table>
もとの属性に戻すときは ESC [m を使用します。
ただし、パラメータ ps は一度に複数指定可能ですが、パラメータ 1 または 7 は、パラメータ30～37と40～47を次のように変化させるトグルスイッチ（実行する度に 2 つのパラメータを切りかえる）の役割を果たします。</td></tr>
</table>

シーケンス	機　　能
	1　　　　　7 30 ↔ 34　　30 ↔ 40 31 ↔ 35　　31 ↔ 41 32 ↔ 36　　32 ↔ 42 33 ↔ 37　　33 ↔ 43 40 ↔ 44　　34 ↔ 44 41 ↔ 45　　35 ↔ 45 42 ↔ 46　　36 ↔ 46 43 ↔ 47　　37 ↔ 47

付録3　テレビコントロールコード一覧

パラメータ	状　　態
$01	ボリュームを上げる
$02	ボリュームを下げる
$03	ボリュームをふつうにする
$04	チャンネルコール
Tv	テレビ画面(初期化・リセット)
$06	音声ミュート
$07	電源ON
$08	テレビ／コンピュータ
$09	テレビ／外部、コンピュータノーマル／オーバー
$0A	コントラストノーマル
$0B	チャンネルアップ
$0C	チャンネルダウン
Off	電源OFF
$0E	電源ON／OFF
$0F	スーパー1
$10	チャンネル1
$11	チャンネル2
$12	チャンネル3
$13	チャンネル4
$14	チャンネル5
$15	チャンネル6
$16	チャンネル7
$17	チャンネル8
$18	チャンネル9
$19	チャンネル10
$1A	チャンネル11
$1B	チャンネル12
$1C	テレビ画面($05)
$1D	コンピュータ画面($05＋$08)
$1E	スーパー1($05＋$0F)
$1F	スーパー2($05＋$0F＋$0A)

※$20～$3Fのとき、電源をONにしたのち、$01～$1F($20＋$01～$20＋$1F)の処理を行います。

索　引

記号類

? （ワイルドカード） …… 24

| （縦線） …… (3)

| （パイプ） …… 43

| | （マルチ処理） …… 44

…… （省略記号） …… (3)

" （ダブルクォーテーション） …… 36

[ ] （角形カッコ） …… (3)

{ } （波型カッコ） …… (3)

〈 〉 （山形カッコ） …… (3)

＜ （入力のリダイレクト） …… 43

＞ （出力のリダイレクト） …… 42

＞＞ （アペンドのリダイレクト） …… 42

¥ （パス名の区切り記号） …… 30

¥ （ルートディレクトリ） …… 30

%0～%9（仮パラメータ） …… 40

＊ （ワイルドカード） …… 24

→・・・（タブ記号） …… 288

↓ （改行記号） …… 288,292

16進表現・文字によるファイル内容の表示（→DUMP） …… 90

1画面ごとの表示（→MORE） …… 116

A～Z

ASCII制御コード …… 301

AUTOEXEC.BAT …… 16,41

AUX（補助入出力装置） …… 25

BELL …… 226

BREAK …… 226

BUFFERS …… 227

Cl …… 190

CA …… 190

CACHE …… 56

CLOCK …… 25

COMMANDコマンドのエラーコード …… 62

COMMONファンクションで使用するメモリ容量の指定（→COMMON） …… 227

CON（コンソール） …… 25

CONFIG.SYS …… 223,232

CONFIG.SYSのコマンド …… 223

CONFIG.SYSファイルの作成・修正（→CONFIGED） …… 232

CTRL+C …… 37,55,186

CTRL+S …… 37

CTRL+Z …… 39

CTRL+Cのチェック機能の設定（→BREAK） …… 55

CTRL系のコマンド …… 199,279

CU …… 190

d：（ドライブ名） …… 36

DEVICE …… 228

DIRSCH …… 231

ECHOモード …… 181

ED …… 275

ED.$$$ …… 275

EDの起動 …… 275

EDのコマンド一覧 …… 296

EDの終了 …… 277

EDのスイッチ付き起動 …… 292

ENVSET …… 231

EOF …… 190,289,292

ESC系のコマンド …… 199,278

EXCONFIG …… 232

FASTIO …… 92

FASTOPEN …… 93

FASTSEEK …… 94

FDDEVICE …… 241

FILES …… 235

FLOAT2.X …… 229

FLOAT3.X …… 229

FLOAT4.X …… 229

FM音源 …… 245

HISTORY.X …… 199

Human68k …… 3,9

Human68k ver.3.0 …… 7

Human68kシステムの転送（→SYS） … 161
Human68kのコマンド …………………… 45
INS……………………………………190
IOCS.X ……………………………………7
KEY …………………………… 110,235
KEY.SYS………………………………235
LASTDRIVE ……………………………236
LPT（プリンタ） ……………………… 25
MIDI………………………………………7
MMLデータ ………………………245,247
N&CU ……………………………… 190
NUL（ヌルファイル） ………………… 25
NWL………………………………………190
OPMDRV3.X ………………………… 229
OPMコマンド ………………………… 254
OPM制御 ……………………………… 252
OPMドライバ ………………………… 242
OPMファイル ………………………… 244
OS ……………………………………… 3
PATH…………………………………37,118
PCM（PCMデバイス） …………………25
PCMファイル ………………………… 226
PRN（プリンタ） ……………………… 25
PRNDRV.SYS ………………………… 267
PRNDRV1.SYS ………………………… 267
PRNDRV2.SYS ………………………… 267
PRNDRV3.SYS ………………………… 267
PROCESS…………………………122,236
PROGRAM ……………………………237
RAMDISK.SYS …………………………269
RAMディスク ………………………… 268
RS-232C…………………………………140
RS-232Cインターフェイスに対する
パラメータの設定・起動（→SPEED） … 140
RS-232C回線 ………………………… 74
S1 ……………………………………… 190
SCSIDEV ………………………………237
SHARE……………………………………238
SHELL…………………………… 224,238
SRAMDISK.SYS ……………………… 269
SU ………………………………………190
TITLE ……………………………………239
TITLE.SYS………………………………239
USKCG……………………………………239
USKCG.SYS …………………………… 239
VERIFY ………………………………… 240
VOID………………………………………190

あ

アスキー比較 ……………………………… 95
新しいディレクトリの作成（→MKDIR）
……………………………………………… 115
アドレス固定実行ファイル…………………24
アペンドのリダイレクト……………………42
移動（ディレクトリの） …………………32
上書…………………………………………280
上書モード…………………………………280
エイリアス（別名定義機能） ……… 198,207
エスケープシーケンス …………………… 303
エラーコード（COMMAMDコマンドの）
………………………………………………62
オーバレイ ………………………………… 52
オーバレイXファイルの作成・変更
（→BIND） ……………………………… 52
大文字変換…………………………………287
大文字変換モード …………………………288
オプション ………………………………… 36
オペレーティングシステム ……………… 3
親ディレクトリ …………………………… 32
親プロセス ………………………………… 91
音源の制御…………………………………245

か

カーソルの移動……………………… 200,278
改行記号……………………………… 288,292
階層構造 …………………………………… 28
階層ディレクトリ構造………………………28
角形カッコ ……………………………… (3)
拡張子 ……………………………………… 23
仮想ドライブ ……………………………… 88

仮想ドライブの割り当ての設定・解除
(→SUBST) …… 144
仮想ドライブを含めた最大のドライブ数を指定
(→LASTDRIVE) …… 236
カッコ …… (3)
カットバッファ …… 286
仮パラメータ …… 40,188
カレントディレクトリ …… 32
カレントディレクトリの変更・表示
(→CHDIR) …… 57
カレントドライブ …… 10
環境エリア …… 137
環境エリアサイズ …… 61
環境文字列 …… 137
環境文字列の値の設定 (→SET) …… 137
外字登録用データファイルの指定
(→USKCG) …… 239
外字の作成・登録・削除
(→USKCGM) …… 170,174
外字ファイル …… 170
外字ファイルの更新 …… 170
外字ファイルの登録 …… 170
外字ファイルの内容 …… 174
外部コマンド …… 35,37
外部コマンド検索のためのPATHの設定
(→PATH) …… 118
画面の消去 (→CLS) …… 60
画面モードの指定 (EDの) …… 293
画面モードの設定 (→SCREEN) …… 135
簡易バッチ機能 …… 199,209
キー設定ファイル …… 110,113
キー設定ファイルの更新 …… 110
キー設定ファイルの内容 …… 113
キー定義ファイル …… 211,235
キー定義ファイルの指定 (→KEY) …… 235
キーの登録 …… 112
キーボードコントロール機能 …… 222
木構造 …… 28
起動 (ED、スイッチ付き) …… 292
起動 (EDの) …… 275
起動 (Human68kの) …… 9
起動 (システムの) …… 9
起動 (内蔵ハードディスクの) …… 20
行長の指定 …… 292
区切り記号 …… 37
検索 …… 281
コピー (フロッピーディスクの) …… 14
コピー (ファイルの) …… 26
子プロセス …… 61,91
子プロセスとして起動されたCOMMAND.X
からの親プロセスへのリターン (→EXIT) …… 91
コマンド …… 35
コマンド (Human68kの) …… 45
コマンドオプション …… 36
コマンド行 …… 181
コマンドの機能別分類 …… 45
コマンドの書式 …… (3)
コマンドの反復実行 (→FOR) …… 183
コマンドの表記法 …… (3)
コマンドプロセッサ …… 231
コマンドプロセッサCOMMAND.Xの起動
(→COMMAND) …… 61
コマンドモード …… 9,10
小文字変換 …… 287
小文字変換モード …… 288

さ

再実行 〈R〉 …… 270
再編集 …… 287
削除 …… 281
削除 (ファイルの) …… 27
削除 (ディレクトリの) …… 33
作成 (ディレクトリの) …… 32
サブディレクトリ …… 28
シェル …… 238
時刻の表示・アラームのセット
(→TIMER) …… 165
シェル起動の前に実行するプログラムの指定
(→PROGRAM) …… 237
システム構築用ファイル …… 223

システムの起動……………………………… 9
システムの終了 ………………………… 11
システム、辞書ディスクの転送 …………… 20
システムのバージョンの表示(→VER) … 175
システムファイル ………………………… 161
実ドライブ………………………………… 144
指定（ファイルの）………………………29
終了（EDの）………………………… 277
終了（システムの）………………………11
終了（編集の）………………………… 289
終端記号（ファイルの）…………… 289,292
出力のリダイレクト………………………42
主ファイル名 ………………………… 23
省略記号 …………………………………(3)
使用可能なメモリサイズの表示
(→MEMFREE) …………………………114
初期化………………………………17,99
初期化（フロッピーディスクの）………… 11
書式（コマンドの）………………………45
時刻の表示・設定（→TIME）………… 164
実パラメータ ……………………… 40,188
自動実行バッチファイル（AUTOEXEC.BAT）
……………………………………………16,41
条件設定によるバッチ処理（→IF）……… 185
スイッチ（コマンドオプション）……………36
スイッチ（メモリスイッチの設定）……… 146
水平タブ …………………………… 288,292
スクリーンエディタ …………………… 275
スクロール ……………………………… 37
セーブ（ファイルの）……………………289
絶対指定 ……………………… 31,57,115
ソート ……………………………………139
ソートキー ………………………………139
相対指定 ……………………… 31,57,115
挿入………………………………………280
挿入モード ………………………………280
ソフトリロケータブル実行ファイル…………24

た

タイトルファイルの指定（→TITLE）……239
退避（ハードディスク上のファイルをフロッピーディスクへ）（→COPY2）………………… 68
縦線 …………………………………………(3)
タブ記号 ……………………………288,293
ダブルクォーテーション ………………… 36
単独キー系のコマンド……………………278
置換…………………………………………282
中止〈A〉 ………………………………… 270
注目行………………………………………196
ツリー構造 ……………………………… 28
データ …………………………………… 16
データのソート（→SORT） ……………… 139
データの並べ変え（→SORT） ………… 139
データファイル……………………………26
ディスク全体のコピー・比較照合
(→DISKCOPY) ……………………………82
ディスクの状態の報告
(→CHKDSK) ………………………… 58
ディスクの初期化（→FORMAT）…………99
ディスクのフォーマット（→FORMAT）… 99
ディスクのボリュームラベルの表示・変更
(→VOL) ……………………………… 177
ディスクバッファ数 …………………… 227
ディスプレイの消去（→CLS）………… 60
ディレクトリ………………………………28
ディレクトリ構造のツリー形式での表示
(→TREE) ……………………………168
ディレクトリの移動……………………… 32
ディレクトリの削除……………………… 33
ディレクトリの削除（→RMDIR） ……… 134
ディレクトリの作成 ………………… 32,115
ディレクトリ（ファイル名一覧）の表示
(→DIR) ………………………………… 79
ディレクトリ変更履歴……………… 198,206
テキストの移動 ………………………… 286
テキストの複写 ………………………… 287
テキストファイル…………………………26
デバイスエラー ………………………… 270
デバイスドライバ…………………… 223,228
デリミタ ………………………………… 37

テンプレート …… 189
テンプレート機能 …… 189,202
同時オープン（ファイルの） …… 235
ドライブの種類、状態の表示・ドライブ名の交換（→DRIVE） …… 88

な

内蔵ハードディスクの初期化 …… 17
内部コマンド …… 35
波形カッコ …… (3)
入出力デバイスの変更（→CTTY） …… 74
入力のリダイレクト …… 43

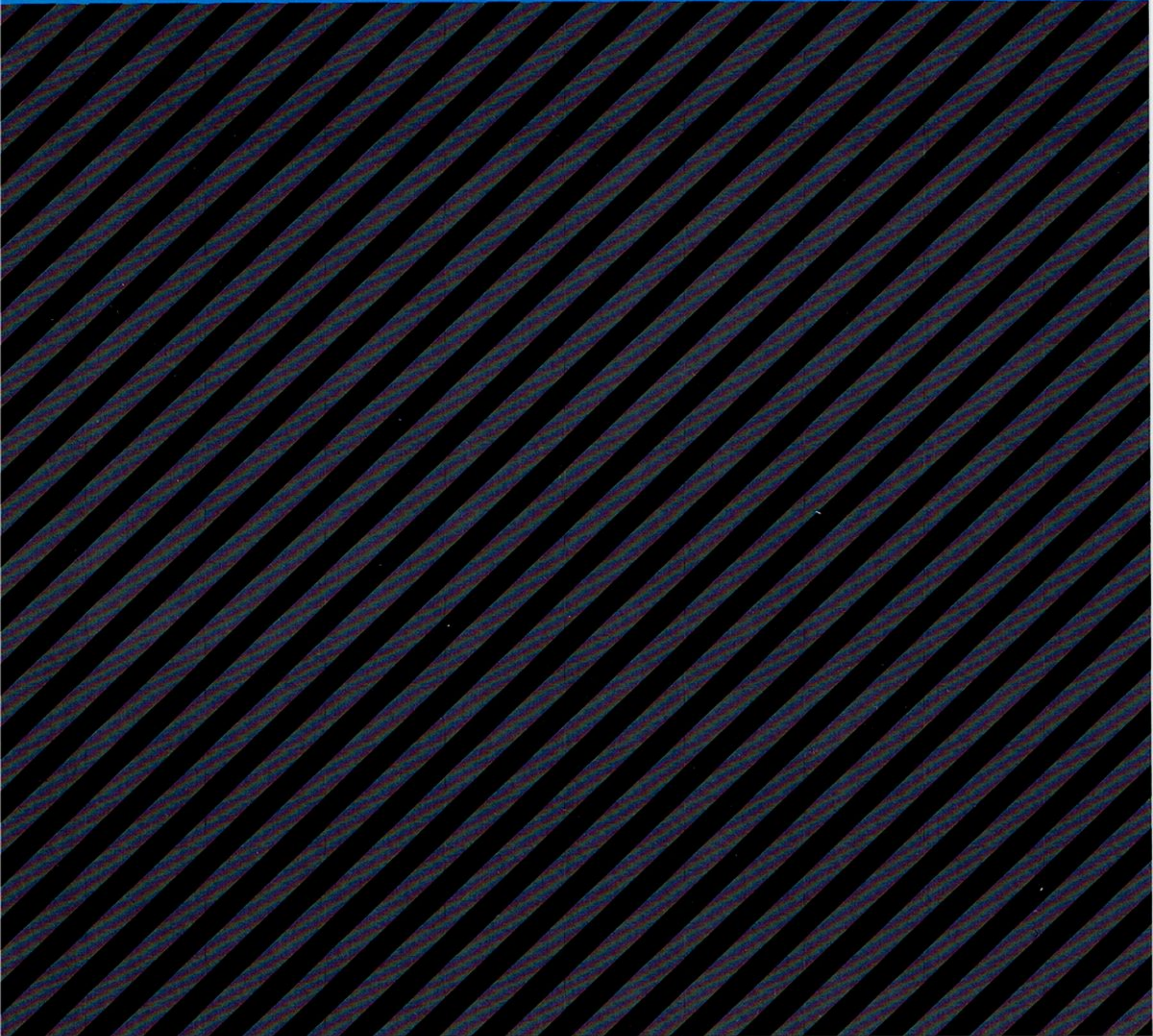
は

ハードディスクからフロッピーディスクへのファイルのバックアップ作成（→BACKUP） …… 50
ハードディスク間のコピー・比較照合 …… 86
ハードディスクのフォーマット …… 101
バイナリ比較 …… 95
バイナリファイル …… 90
パイプ機能 …… 43
パス …… 28
パス名 …… 30,118
バックアップ（フロッピーディスクの） …… 14
バックアップ（ハードディスクの） …… 50
BACKUPコマンドでバックアップされたファイルの復元（→RESTORE） …… 132
バッチ処理 …… 38
バッチ処理コマンド …… 180
バッチ処理中におけるコマンド行表示の設定・メッセージの表示（→ECHO） …… 181
バッチ処理中におけるコメントの表示（→REM） …… 187
バッチ処理中におけるパラメータのシフト（→SHIFT） …… 188
バッチ処理の一時停止（→PAUSE） …… 186
バッチ処理の流れの変更（→GOTO） …… 184
バッチファイル …… 38
パラメータ …… 36
パラメータ（バッチファイルの） …… 40
範囲指定 …… 286
ビープ音 …… 226
ビジュアルモード …… 10
ヒストリエリアサイズ …… 62
ヒストリ機能 …… 194,197,203
ヒストリ行列 …… 108,194
ヒストリ行列の表示（→HIS） …… 108
ヒストリデバイスドライバ …… 197
ヒストリ定義ファイル …… 211
日付の表示・設定（→DATE） …… 75
表記法（コマンドの） …… (3)
標準出力 …… 42
標準入力 …… 42
ファイル …… 16,23
ファイルおよびディレクトリのコピー（→COPYALL） …… 72
ファイル終端記号 …… 289,292
ファイル内容の比較（→FC） …… 95
ファイル内容の表示（→TYPE） …… 169
ファイル内容のページング（→PR） …… 120
ファイルの移動（→MOVE） …… 117
ファイルの書き出し …… 285
ファイルの検索（→WHERE） …… 179
ファイル中の指定文字列の検索（→FIND） …… 97
ファイルの共有と排他制御の指定（→SHARE） …… 238
ファイルのコピー …… 26
ファイルのコピー（→COPY） …… 64
ファイルの削除 …… 27
ファイルの削除（→DEL） …… 77
ファイルの指定 …… 29
ファイルのセーブ …… 289
ファイルの属性の設定・解除（→ATTRIB） 48
ファイルの読み込み …… 285
ファイルハンドラ …… 235
ファイル名 …… 23
ファイル名の付け方 …… 23
ファイル名の変更（→REN） …… 130
ファンクションキーなどの設定（→KEY） 110
フィルタ …… 43,98,116,120,139

フォーマット……………………………………11
不可視属性 …………………………………… 48
複数ファイルの編集 ………………………… 290
復帰（フロッピーディスク上の退避ファイルをハードディスクへ）（→COPY2）………………68
不良セクタを含むファイルまたはディスクの修復（→RECOVER） ………………………… 127
フルパス名……………………………………31,115
プログラムファイル ………………………… 26
プロセス ……………………………………… 122
プロセスの情報の表示（→PROCESS） … 122
フロッピーディスクのフォーマット…………99
プロンプト …………………………………… 10
プロンプトの設定（→PROMPT） ……… 125
並行処理のための制御情報の設定（→PROCESS） ……………………………236
ベリファイ機能の設定（→VERIFY） …… 176
ヘルプ画面（→ATTRIB） ………………… 49
ヘルプファイル ……………………… 221,292
編集の終了 …………………………………… 289
編集ファイルの切りかえ……………………290
ボーレート …………………………………… 141
補助入出力機器………………………………74
補助入出力機器（→AUX） ………………… 25
ボリュームラベル ………………………… 177

ま

マルチ処理機能………………………………44
無視〈I〉 ……………………………………270
メモリスイッチの設定（→SWITCH） …… 146

や

山形カッコ ………………………………… (3)
横スクロール ……………………………… 276
読み出し専用属性…………………………… 48
予約ファイル名………………………………25

ら

ラベル……………………………………… 184
リターン ………………………………… (3)
リダイレクト機能…………………………… 42
領域解放…………………………………17,104
領域確保 …………………………………… 17
リロケータブル実行ファイル ……………… 24
ルートディレクトリ………………………28,30

わ

ワイルドカード………………………………24

シャープ株式会社

本　　社　〒545 大阪市阿倍野区長池町22番22号　電話(06) 621-1221(大代表)

電子機器事業本部　〒329-21 栃木県矢板市早川町174番地　電話(0287)43-1131(大代表)

TMAN-3737CEZZ

T0593-A ②